

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV VÝKONOVÉ ELEKTROTECHNIKY A ELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF POWER ELECTRICAL AND ELECTRONIC ENGINEERING

## **ANALÝZA STEJNOSMĚRNÉHO STROJE POMOCÍ PROGRAMU LabVIEW**

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

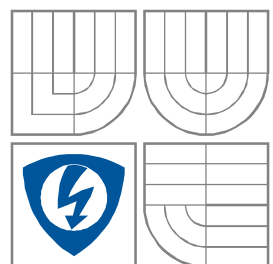
Lukáš Loun

BRNO 2008



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLÓGIÍ**

**ÚSTAV VÝKONOVÉ ELEKTROTECHNIKY  
A ELEKTRONIKY**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF RADIO ELECTRONICS

# **ANALÝZA STEJNOSMĚRNÉHO STROJE POMOCÍ PROGRAMU LabVIEW**

**DC MACHINE MEASUREMENT AND ANALYSIS USING LABVIEW**

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

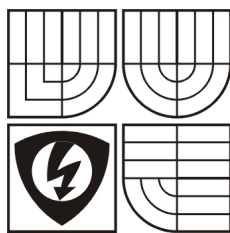
Lukáš Loun

**VEDOUCÍ PRÁCE**

SUPERVISOR

Ing. Ondřej Vítek, Ph.D.

BRNO, 2008



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav výkonové elektrotechniky a elektroniky

# Diplomová práce

magisterský studijní obor  
Silnoproudá elektrotechnika a výkonová elektronika

**Student:** Lukáš Loun

**Ročník:** 2

**ID:** 89536

**Akademický rok:** 2007/08

**NÁZEV TÉMATU:**

**Analýza stejnosměrného stroje pomocí programu LabView**

**POKYNY PRO VYPRACOVÁNÍ:**

1. Analyzujte možnosti virtuálního měření laboratorních úloh v elektrotechnice.
2. Popište konstrukci a princip funkce stejnosměrného motoru a vytvořte podklady pro jeho měření.
3. Realizujte laboratorní úlohu na automatizované měření stejnosměrných strojů pomocí programu LabView.

**DOPORUČENÁ LITERATURA:**

**Termín zadání:** 10.10.2007

**Termín odevzdání:** 30.5.2008

**Vedoucí projektu:** Ing. Ondřej Vítek, Ph.D.

**doc. Ing. Čestmír Ondrůšek, CSc.**  
*předseda oborové rady*

**UPOZORNĚNÍ:**

Autor semestrální práce nesmí při vytváření semestrální práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

# LICENČNÍ SMLOUVA

## POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

### 1. Pan/paní

Jméno a příjmení: Lukáš Loun

Bytem: Hranická 21, Přerov 75124

Narozen/a (datum a místo): 19.11.1983 v Přerově

(dále jen „autor“)

a

### 2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií

se sídlem Údolní 53, Brno, 602 00

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

doc. Ing. Čestmír Ondrůšek, CSc., předseda oborové rady Silnoproudá

elektrotechnika a elektroenergetika

(dále jen „nabyvatel“)

## Čl. 1

### Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

☐ disertační práce

☒ diplomová práce

☐ bakalářská práce

☐ jiná práce, jejíž druh je specifikován jako .....

(dále jen VŠKP nebo dílo)

Název VŠKP:

Vedoucí/ školitel VŠKP:

Ústav:

Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v \*:

☒ tištěné formě – počet exemplářů 1

☒ elektronické formě – počet exemplářů 1

---

\* hodící se zaškrtněte

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

## **Článek 2**

### **Udělení licenčního oprávnění**

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
  - ☐ ihned po uzavření této smlouvy
  - ☐ 1 rok po uzavření této smlouvy
  - ☐ 3 roky po uzavření této smlouvy
  - ☐ 5 let po uzavření této smlouvy
  - ☐ 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## **Článek 3**

### **Závěrečná ustanovení**

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....

Nabyvatel

.....

Autor

## **Abstrakt**

Virtuální laboratoře se dělí do dvou skupin. První jsou virtuální laboratoře sloužící k simulaci. Většinou se používají k prezentačním a demonstračním účelům, k jejich použití není třeba instalovat složitý a drahý software.

Další skupinou virtuálních laboratoří jsou tzv. dálkové laboratoře. Pomocí tohoto druhu laboratoří se měří skutečné systémy. Výhodou dálkového měření je, že data z měření jsou přímo ukládána do počítače a mohou být okamžitě vyhodnocována. V práci je popsán postup při návrhu a realizaci virtuálního přístroje pro dálkové měření v programu LabVIEW.

LabVIEW je vývojovým grafickým prostředím zaměřeným na vývoj aplikací umožňujících měření, sbírání dat a analýzu.

## **Abstract**

The virtual laboratory is divided into two groups. The first group is comprised of a virtual laboratory for simulation. It is mostly used for presentation and demonstration purposes. It is not necessary to install complicated and expensive software for their use.

The second group of the virtual laboratory is a long distance measure. Real systems are measured with the help of this laboratory. The advantage of long distance measuring is that data from the measurements are saved directly onto the personal computer and they can be immediately evaluated. In this project the procedure for creating a virtual instrument for long distance measuring is described in the LabVIEW program.

LabVIEW is a graphic developmental environment, for the measuring and analysis of data.

### **Klíčová slova**

virtuální laboratoř; automatizované měření; vzdálená laboratoř; stejnosměrný motor; LabVIEW

### **Keywords**

Virtual laboratory; automated measuring; long distance laboratory; DC motor; LabVIEW

## **Bibliografická citace**

LOUN, L. *Analýza stejnosměrného stroje pomocí programu LabView*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 69 s. Vedoucí diplomové práce Ing. Ondřej Vítek, Ph.D.



## **Prohlášení**

Prohlašuji, že svou diplomovou práci na téma Analýza stejnosměrného stroje pomocí programu LabView jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne .....

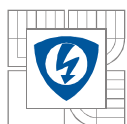
Podpis autora .....

## **Poděkování**

Děkuji vedoucímu diplomové práce Ing. Ondřeji Vítkovi za účinnou metodickou, pedagogickou a odbornou pomoc a dalším za cenné rady při zpracování mé diplomové práce.

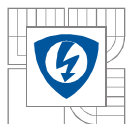
V Brně dne .....

Podpis autora .....



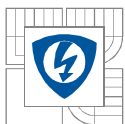
## OBSAH

<b>1 ÚVOD .....</b>	<b>14</b>
<b>2 VIRTUÁLNÍ LABORATOŘE.....</b>	<b>15</b>
<b>3 PRINCIP STEJNOSMĚRNÉHO MOTORU .....</b>	<b>16</b>
<b>3.1 STEJNOSMĚRNÝ MOTOR SE SÉRIOVÝM BUZENÍM .....</b>	<b>18</b>
<b>4 PROGRAMOVACÍ JAZYK LABVIEW .....</b>	<b>21</b>
<b>4.1 PROGRAMOVÁNÍ V LABVIEW .....</b>	<b>22</b>
<b>4.2 KOMUNIKACE S HW .....</b>	<b>23</b>
4.2.1 PARAMETRY MĚŘICÍ KARTY USB-6251.....	24
4.2.2 ZPRACOVÁNÍ DAT .....	24
4.2.3 POŽADAVKY OS.....	25
<b>4.3 UKLÁDÁNÍ DO SOUBORŮ .....</b>	<b>25</b>
<b>5 NÁVRH A REALIZACE VIRTUÁLNÍ LABORATOŘE .....</b>	<b>26</b>
<b>5.1 TVORBA VIRTUÁLNÍHO PŘÍSTROJE.....</b>	<b>27</b>
<b>5.2 AUTOMATIZOVANÉ MĚŘENÍ PRO STEJNOSMĚRNÝ MOTOR.....</b>	<b>32</b>
5.2.1 NASTAVENÍ DAQ ASSISTENTA.....	34
5.2.2 PROGRAMOVATELNÝ ZDROJ HEINZINGER .....	37
5.2.3 HIERARCHICKÁ STAVBA – TVOŘENÍ SUB VI.....	41
5.2.4 REPORTOVÁNÍ - MOŽNOSTI UKLÁDÁNÍ NAMĚŘENÝCH VELIČIN A GRAFŮ .....	43
5.2.5 VYTVOŘENÍ TABULKY .....	46
5.2.6 GRAFY .....	48
5.2.7 OSCILOSKOP.....	48
5.2.8 PŘEVOD MĚŘENÝCH VELIČIN.....	49
5.2.9 VYTVOŘENÍ INSTALAČNÍHO SOUBORU .....	49
<b>5.3 POSTUP MĚŘENÍ .....</b>	<b>50</b>
<b>5.4 POUŽITÉ PŘÍSTROJE V MĚŘICÍM SYSTÉMU: .....</b>	<b>53</b>
<b>6 ZHODNOCENÍ .....</b>	<b>54</b>
<b>LITERATURA .....</b>	<b>55</b>
<b>PŘÍLOHY.....</b>	<b>56</b>



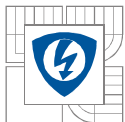
## SEZNAM OBRÁZKŮ

<i>Obr. 3.1 Smyčka s komutátorem, rotující v homogenním magnetickém poli .....</i>	<i>16</i>
<i>Obr. 3.2 Časový průběh točivého momentu .....</i>	<i>17</i>
<i>Obr. 3.3 Schéma zapojení sériového motoru .....</i>	<i>18</i>
<i>Obr. 3.4 Závislost veličin <math>P</math>, <math>M</math>, <math>I</math>, <math>\Phi</math> na rychlosti <math>n</math> sériového motoru .....</i>	<i>20</i>
<i>Obr. 4.1 A/D převodník, měřicí karta NI USB-6251 .....</i>	<i>24</i>
<i>Obr. 5.1 Blokové schéma měřicího systému pro ss motor.....</i>	<i>26</i>
<i>Obr. 5.1.1 Úvodní obrazovka pro tvorbu projektu.....</i>	<i>27</i>
<i>Obr. 5.1.2 Okno vyhledávání a správu příkladů .....</i>	<i>28</i>
<i>Obr. 5.1.3 Okno Project Explorer.....</i>	<i>28</i>
<i>Obr. 5.1.4 Front Panel – vnější vzhled přístroje.....</i>	<i>29</i>
<i>Obr. 5.1.5 Block Diagram.....</i>	<i>30</i>
<i>Obr. 5.1.6 Controls panel pro tvorbu ve Front Panelu.....</i>	<i>31</i>
<i>Obr. 5.1.7 Function panel pro tvorbu virtuálních přístrojů v okně Block Diagram.....</i>	<i>32</i>
<i>Obr. 5.1.8 Context Help - Červeně je zvýrazněn kurzor myši a okno kontextové nápovědy.....</i>	<i>33</i>
<i>Obr. 5.1.9 Možnosti zobrazení bloků .....</i>	<i>33</i>
<i>Obr. 5.1.10 DAQ Assistant.....</i>	<i>34</i>
<i>Obr. 5.1.11 Druhy zapojení měřicí karty .....</i>	<i>35</i>
<i>Obr. 5.1.12 Blokové schéma měření napětí.....</i>	<i>36</i>
<i>Obr. 5.1.13 Nastavení vstupů na měřicí kartě National Instruments USB-6251 .....</i>	<i>37</i>
<i>Obr. 5.1.14 Přístrojový model SCPI .....</i>	<i>38</i>
<i>Obr. 5.1.15 USB/GPIB převodník Agilent .....</i>	<i>39</i>
<i>Obr. 5.1.16 Schéma řízení zdroje Heinzinger .....</i>	<i>40</i>
<i>Obr. 5.1.17 Schéma subVI v nadřazeném VI.....</i>	<i>42</i>
<i>Obr. 5.1.18 Detailní náhled na subVI zobrazené v Obr. 23.....</i>	<i>42</i>
<i>Obr. 5.1.19 DIAdem Report blok .....</i>	<i>43</i>
<i>Obr. 5.1.20 Schéma pro reportování do HTML.....</i>	<i>44</i>
<i>Obr. 5.1.21 Varovné okno pro změnu záložky.....</i>	<i>45</i>
<i>Obr. 5.1.22 Ukázka ukládání protokolu ve Front Panelu.....</i>	<i>45</i>
<i>Obr. 5.1.23 Část programu zajišťující ukládání řádku do tabulky .....</i>	<i>47</i>
<i>Obr. 5.1.24 Uložení tabulky do Excelu .....</i>	<i>47</i>
<i>Obr. 5.1.25 Výběr signálů pro osciloskop.....</i>	<i>49</i>



## SEZNAM SYMBOLŮ A ZKRATEK

$P_m$	mechanický výkon motoru	$W$
$M$	moment	$N \cdot m$
$\omega$	mechanická úhlová rychlost	$s^{-1}$
$P_e$	elektrický výkon	$W$
$u_i$	indukovaná napětí	$V$
$I$	elektrický proud	$A$
$n$	otáčky	$ot \cdot min^{-1}$
$f$	kmitočet	$Hz$
$T$	perioda	$s$
$p$	počet pólů	-
$\phi$	magnetický tok	$Wb$
$U$	napětí	$V$
$M_m$	točivý moment	$N \cdot m$
$P$	výkon	$W$
$P_p$	příkon	$W$
$P_{Fe}$	ztráty v železe	$W$
$P_{jR}$	ztráty v rotoru (kotvě)	$W$
$P_m$	mechanické ztráty	$W$



## **SEZNAM PŘÍLOH**

### **PŘÍLOHA 1**

Zatěžovací charakteristika spouštěče

### **PŘÍLOHA 2**

Charakteristika spouštěče naprázdno

### **PŘÍLOHA 3**

Charakteristika spouštěče nakrátko

### **PŘÍLOHA 4**

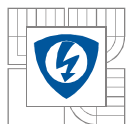
Ovládací panel programu EasyMot

### **PŘÍLOHA 5**

Blokové schéma programu EasyMot

### **PŘÍLOHA 6**

Bloková schémata SubVI použité v programu EasyMot



# 1 ÚVOD

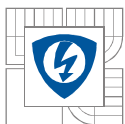
Virtuální laboratoře zažívají v posledních letech obrovský vzestup. Asi největší podíl na vývoji virtuálních laboratoří má rozvoj a zpřístupnění internetu. V dnešní době patří internet ke standardní výbavě domácnosti stejně jako televize, na internetu jsou banky, obchody, dokonce i lékaři, tak proč by na internetu nemohla existovat i laboratoř. Bez nadsázky lze říci, že internet se stal hlavním zdrojem informací, který se stále rozvíjí a není to jenom svět textových informací, ale dynamický systém, přes který je možno provádět měření nebo řídit systém vzdálený tisíce kilometrů. Díky dálkově řízeným laboratořím se dá například rozsvítit žárovka na druhé straně zeměkoule, nebo řídit družice v kosmu.

Virtuální laboratoře se dělí do dvou skupin. První jsou virtuální laboratoře sloužící k simulaci. Většinou se používají k prezentačním a demonstračním účelům, bývají psány v jazycích, které lze spouštět přes webové prohlížeče (Java, Flash) a bez nutnosti instalovat složitý a drahý software. Výhodou tohoto druhu laboratoří je možnost měření (simulace) v jakoukoliv dobu a pro neomezený počet uživatelů zároveň. Uživatel může zadávat vstupní hodnoty stejně jako u reálného měření a přes webový prohlížeč sledovat chování stroje (systému), který se chová jako skutečný.

Další skupinou virtuálních laboratoří jsou tzv. dálkové laboratoře. Pomocí tohoto druhu laboratoří se měří skutečné systémy, jsou přesnější, jelikož se snímají skutečné veličiny. Výhodou dálkového měření je, že data z měření jsou přímo ukládána do počítače a mohou být okamžitě vyhodnocována. Tento postup měření vede k jeho zautomatizování.

Práce se zabývá dálkově řízenými laboratořemi a především dálkovému řízení stejnosměrného motoru. Dálkové řízení je realizováno pomocí grafického programového prostředí LabVIEW od firmy National Instruments. Výsledkem práce je funkční model pro dálkové řízení stejnosměrného motoru se sériovým buzením pomocí počítače.

V tomto textu je nastíněno, že mnoho měření lze provést i bez nutnosti vlastnit drahé měřicí přístroje, ale stačí měřicí karta a sofistikovaný software, ve kterém je možno naprogramovat téměř jakýkoliv měřicí přístroj. Jedinou měřicí kartou a osobním počítačem můžeme nahradit spoustu druhů měřicích přístrojů, což je zajímavým řešením pro mnoho podniků a škol.



## 2 VIRTUÁLNÍ LABORATOŘE

Virtuální laboratoř umožňuje provádět počítačové simulace, u kterých uživatel může interaktivně měnit různé parametry pozorovaných fyzikálních dějů a pracovat s daty, která nemá možnost sám naměřit. Nebo může na dálku měřit skutečné úlohy pomocí speciálního hardwaru a sítě internet nebo ethernet. Mnoho virtuálních laboratoří je vystavěno na základě apletů napsaných v jazyce Java. Přibývá ale i virtuálních laboratoří, které jsou tvořeny animacemi a simulacemi v jazyce Flash. Tato dvě různá programovací prostředí se používají nejvíce, protože nepotřebují k provozování speciální programové vybavení, jako jsou programy LabView, Matlab a podobně, u kterých cena programu dosahuje desítek tisíc. Flash a Java jsou méně náročné a volně šiřitelné programy.

Ať je však programovací jazyk jakýkoliv, základní myšlenka zůstává stejná: aplety a simulace je možné prohlížet pomocí webovského prohlížeče a změnami parametrů demonstrovaného děje zasahovat do jejich průběhu. Toto zasahování může spočívat v zadávání a změnách hodnot fyzikálních veličin (zadáním hodnoty, popotazením ukazatele, atd.), v různém natáčení a zoomování pohledu na demonstrovaný jev, ve zpomalování, zastavování, krokování a vracení časového průběhu a jeho zobrazení.

Virtuální laboratoře, které slouží pouze k simulaci, mají zapsána data v databázích ze skutečného měření. Tudíž veškeré tvorbě virtuálního modelu předchází reálné měření. Výhodou tohoto druhu měření, přesněji simulace, je, že danou úlohu může přes virtuální laboratoř měřit libovolný počet uživatelů. Počet uživatelů je omezen pouze datovým tokem připojení k dané virtuální laboratoři. Oproti tomu k dálkově ovládané laboratoři musí mít přístup pouze jeden aktivní uživatel, protože se jedná o měření reálného stroje, u kterého nemohou ve stejný časový okamžik požadovat dva různí uživatelé dvě odlišná měření. Proto by měl být tento druh měření zabezpečen pouze na jedno síťové připojení, abychom zabránili spuštění více neslučitelných úloh paralelně. U dálkově prováděných měření skutečných strojů musí být měřený objekt zabezpečen proti náhodnému dotyku, nejlépe v uzavřené uzamykatelné místnosti, a to z toho důvodu, aby nedošlo ke zraněním způsobeným dálkovým spuštěním stroje.

Virtuální měření nebo laboratoře jsou v poslední době velmi oblíbenou formou výuky na mnoha školách i univerzitách. Jejich největší výhodou je provádění měření bez nutnosti vlastnit měřený objekt. V tom je také největší potenciál virtuálních laboratoří, protože k provozování virtuální laboratoře stačí pouze osobní počítač s konkrétním programovým vybavením a připojením k internetu. Díky finančně nenáročnému řešení se tento druh měření otevírá i pro širší veřejnost.

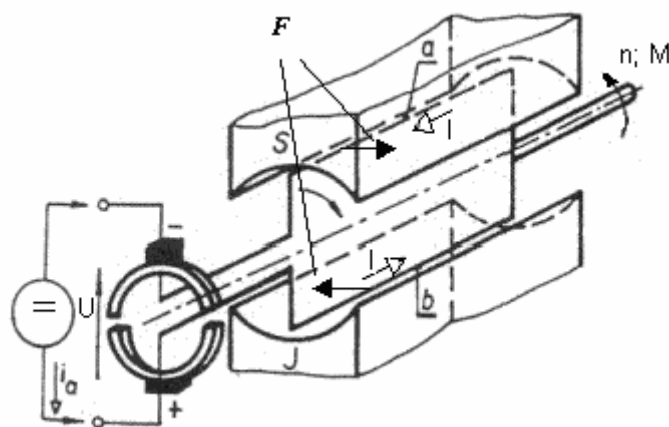
Praktické využití virtuálních a vzdálených laboratoří je např. ve velínech elektráren, tepláren, spaloven, používá se k regulaci vodních toků, ale také ve výrobním průmyslu, ať už ve velkých závodech či menších podnicích. Jeden z důvodů četného použití je v nebezpečných nebo rizikových prostředích, další výhodou je centrální ovládání několika vzdálených pracovišť z jednoho místa.

Použití dálkově řízeného ovládání vede ke snížení celkových nákladů a minimalizaci selhání lidského faktoru, zvýšení spolehlivosti a rychlosti ovládání (měření). V praxi to znamená, že jeden dispečer může řídit několik rozvodů z jedné centrály pomocí sofistikovaného softwaru určeného pro dálkové ovládání, a tudíž není třeba mít na každé rozvodně další osobu pro ovládání. Tímto druhem řízení se i zvyšuje pružnost odezvy na nenadálé situace. Např.: pokud vypadne jedna trafostanice, dispečer může okamžitě reagovat a přesměrovat energii z jiných zdrojů do postiženého místa, a to pouze z jednoho ovládacího bodu, velína.

Virtuálních laboratoří, jakožto simulací, se v elektrotechnice využívá obzvláště k demonstrativním a didaktickým účelům, k popsání určitých dějů a stavů. Kdežto dálkově řízených laboratoří se využívá v mnoha oddílech elektrotechniky, jak už bylo zmíněno výše. Zajímavým příkladem dálkového řízení je například ovládání vesmírných družic a raketoplánu v kosmu z centrály na Zemi.

### 3 PRINCIP STEJNOSMĚRNÉHO MOTORU

Princip činnosti stejnosměrného stroje lze jednoduše vysvětlit na smyčce otáčející se v magnetickém poli. Konce smyčky jsou připojeny ke dvěma segmentům, vzniklým podélným rozříznutím válcového vodiče tvořící komutátor. Toto je nejjednodušší druh komutátoru, v praxi se používá více smyček. Každá smyčka je uzavřena na komutátoru tak, aby její konce byly přesně naproti sobě.



Obr. 3.1 Smyčka s komutátorem, rotující v homogenním magnetickém poli

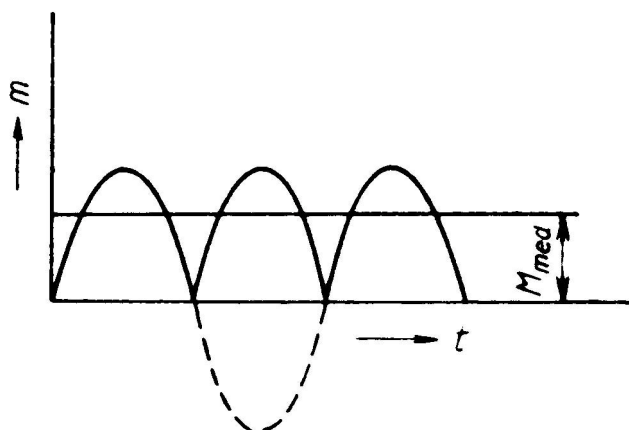
Na komutátor dosedají dva vodivé kartáče, nejčastěji z uhlíku C, připojené přes svorky na zdroj. Zdroj má stejnosměrné napětí  $U$ , které dá vzniknout proudu  $I$ .

Vzniklý proud vytvoří ve smyčce pole, které se spojí s homogenním polem statoru na pole nehomogenní. V jednom místě smyčky dochází ke zhuštění pole, které se chce ve směru kolmém k indukčním čarám rozepnout, a proto působí silou na vodič. Vznikne moment  $M$ , díky kterému se motor roztočí otáčkami  $n$  v jeho směru.



Rychlost  $n$  je určena velikostí indukovaného napětí  $U_i$ , které je až na úbytky stejné jako svorkové napětí  $U$ .

Předpokládáme-li časový průběh proudu nezvlněný, časově konstantní proud, je konstantní i síla  $F$ , ale točivý moment vzhledem k změně úhlu mezi směrem síly a ramenem síly se mění sinusově. Okamžitá hodnota je nulová v místě, kde plocha smyčky je kolmá na směr pole. Dochází ke změně proudu v cívice tzv. komutaci, kde záporná půlvlna sinusovky se překlopí do kladné poloroviny a moment nemění svůj směr. V časovém průměru ukazuje smyčka střední hodnotu točivého momentu  $M_{med}$ .



Obr. 3.2 Časový průběh točivého momentu

Mechanický výkon motoru je

$$P_m = M \cdot \omega \quad (1)$$

kde  $M$  je točivý moment vytvořený magnetickým polem a  $\omega$  mechanická úhlová rychlost.

Podle zákona o zachování energie musí být elektrický výkon motoru přivedený (příkon)

$$P_e = U_i \cdot I \quad (2)$$

stejně velikosti (ztráty neuvažujeme) jako mechanický výkon. Proto musí platit:

$$M \cdot \omega = U_i \cdot I \quad (3)$$

U motoru je smysl otáčení stejný jako směr momentu vznikajícího ve stroji a směr proudu je stejný se směrem napětí  $U_i$ , popř.  $U$ .

Naopak u generátoru je smysl otáčení momentu a otáček opačný, stejně tak je rozdílný i směr napětí a proudu.

U skutečných stejnosměrných strojů se ve vinutí používá mnoho takových smyček, které jsou rovnoměrně uloženy po obvodu, tudíž zvlnění křivky napětí nebo točivého momentu je zanedbatelné.

Magnetické pole není také homogenní, ale je prostorově proměnné a indukční čáry procházejí převážně železem. Indukované napětí vzniká v každé smyčce (cívce) vždy jako střídavé napětí, které se musí usměrnit komutátorem. Jeho časový průběh se periodicky opakuje za dobu kmitu  $T$ , která u dvojpólového stroje odpovídá úhlu  $360^\circ$ , to znamená jedné otáčky rotoru. Tato doba kmitu je rovna převrácené hodnotě počtu otáček za sekundu  $\frac{1}{n}$ .

U více pólových strojů s  $p$  pólovými dvojicemi dojde během jedné otáčky k vzniku  $p$  kmitů střídavého napětí. Označíme-li počet kmitů napětí za sekundu jako kmitočet indukovaného střídavého napětí  $f = \frac{1}{T}$ ,

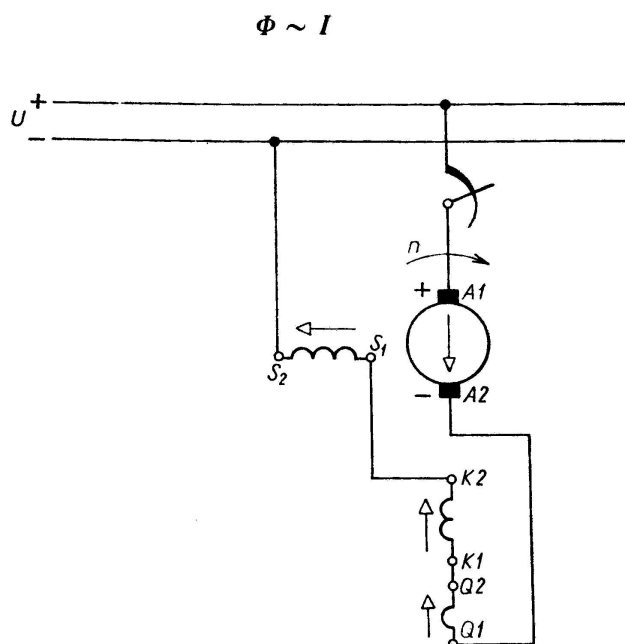
platí vztah

$$f = \frac{1}{T} = p \cdot n \quad (4)$$

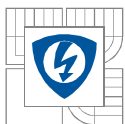
Stejnoseměrné stroje se obvykle vyrábějí jako vícepólové varianty, minimálně čtyřpólové, a to z důvodu lepšího využití stroje.

### 3.1 Stejnoseměrný motor se sériovým buzením

Sériový motor má budící vinutí zapojeno do série s vinutím kotvy, tudíž zatěžovací proud je také proudem budícím. Budící vinutí je realizováno vodičem z několika závitů velkého průměru. Pro vysvětlení vlastností si sériový motor zidealizujeme a zanedbáme úbytky činného napětí.



Obr. 3.3 Schéma zapojení sériového motoru



Budeme předpokládat, že svorkové napětí  $U$  se rovná indukovanému napětí  $U_i$   
 $U \sim U_i$ , stejně tak lineární závislost magnetického toku  $\Phi$  na proudu  $I$

$$\Phi \sim I \quad (5)$$

Obecně platí  $U \sim n\Phi$  (6)

$$M_m \sim \Phi I \quad (7)$$

Napětí  $U$  předpokládáme za konstantní, potom po dosazení do (6) dostaneme

$$\Phi \sim \frac{1}{n} \quad (8)$$

z (5) a (6) vyplývá

$$M_m \sim I^2 \quad (9)$$

nebo za pomoci vztahu (6)

$$M_m \sim \frac{1}{n^2} \quad (10)$$

výkon

$$P = UI \sim I \sim \frac{1}{n} \quad (11)$$

Ze vztahu (9) vyplývá, že točivý moment  $M_m$  roste s druhou mocninou zatěžovacího proudu  $I$ , pokud ovšem magnetický obvod není nasycen

Z této vlastnosti je zřejmé, že při poměrně malém zvětšení proudu dostaneme celkem velký přírůstek točivého momentu. Je-li stroj magneticky nasycen a  $\Phi$  je konstantní, točivý moment roste už jen s proudem, a to lineárně. Proud  $I$ , výkon  $P$  a magnetický tok  $\Phi$  sledují podle rychlosti  $n$  hyperbolickou závislost, přičemž točivý moment  $M_m$  je nepřímo úměrný druhé mocnině rychlosti  $n$ . Z těchto závislostí vyplývá, že pro nulový točivý moment a i nulový výkon vychází nekonečně velká rychlost  $n$  při nulovém proudu  $I$  a magnetickém toku  $\Phi$ . Opačně je tomu u rychlosti, kde při nulové rychlosti  $n = 0$  vychází nekonečně velký záběrný moment a proud. Toto platí pouze teoreticky u zjednodušeného modelu.

U skutečného sériového motoru musíme přihlížet i k odporu obvodu motoru  $R$  a k jeho magnetizační charakteristice. Zatěžovací proud u sériového motoru je současně i proudem magnetizačním. Proto bude platit

$$U_i = U - RI \quad (12)$$

rychlost otáčení určíme z

$$n \sim \frac{U_i}{\Phi} \quad (13)$$

a příkon

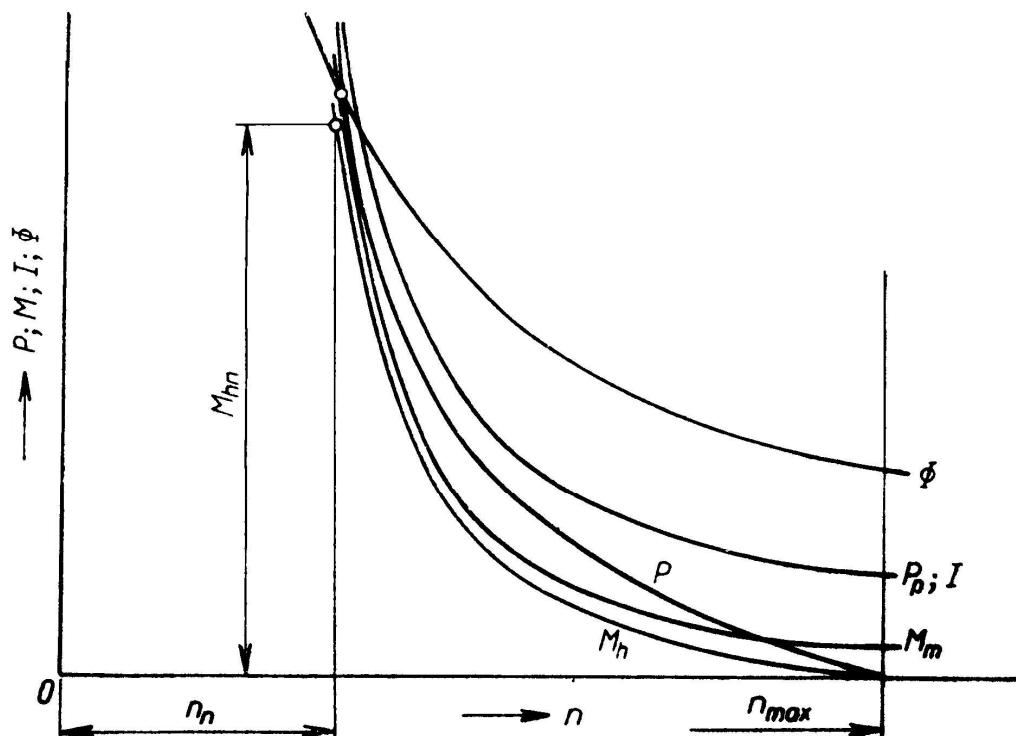
$$P_p = UI \sim I \quad (14)$$

Mechanické ztráty  $\Delta P_m$  nejsou konstantní, při malém proudu  $I$  jsou největší, protože je největší rychlost. Dále ztráty  $\Delta P_{Fe}$  mají při určitém zatížení a rychlosti své minimum, protože pro malé proudy  $I$  jsou velké vzhledem k velkým rychlostem a tedy k vysokému kmitočtu v kotvě. Pro velké proudy  $I$  je sice kmitočet nízký, ale velká je naopak magnetická indukce v železe kotvy. Ztráty  $\Delta P_j$  rostou s druhou mocninou proudu  $I$ . Výstupní výkon na hřídeli  $P$  dostaneme odečtením všech ztrát od příkonu.

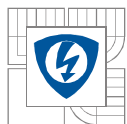
Sériový motor se nesmí připojovat a používat naprázdno, protože by došlo až k nekonečnému nárůstu otáček (teoretický předpoklad) a motor by se zničil. V praxi je tato podmínka splněna tím, že je motor trvale připojen k hřídeli poháněného zařízení.

Sériový motor se spouští stejně jako cize buzený a derivační motor přes spouštěcí předřadné odpory, aby nedošlo při připojení na síť k příliš velkému proudu a k prudkému momentovému rázu.

Momentová charakteristika je závislost momentu na rychlosti  $M(n)$  a sestojí se z charakteristik  $M(I)$  a  $n(I)$ .



Obr. 3.4 Závislost veličin  $P$ ,  $M$ ,  $I$ ,  $\Phi$  na rychlosti  $n$  sériového motoru



Sériové motory dobře vydrží přetížení a spínaný provoz, protože se dokážou samočinně přizpůsobit novým provozním stavům. Pokud dojde k přetížení, motor se dokáže sám přibudit, tím sníží svou rychlost a umožní, aby setrvačné síly přispěly ke krytí špičkových přetížení. Tato vlastnost je nazývána jako měkká nebo také sériová charakteristika.

Sériový motor můžeme analyzovat z jedné veličiny otáček. Známe-li otáčky určitého sériového motoru, známe tím i proud, který odebírá, výkon na hřídeli a i točivý moment. Sériový motor je ideální motor pro elektrickou trakci, jediná jeho nevýhoda je, že s ním nelze rekuperovat do sítě.

## 4 PROGRAMOVACÍ JAZYK LABVIEW

LabVIEW je vývojovým grafickým prostředím využívající programovací prostředí jazyka G. Toto prostředí je zaměřeno na vývoj aplikací umožňujících měření a sbírání dat, analýzu a prezentaci dat.

LabVIEW je zkratkou **L**aboratory **V**irtual **I**nstrumentation **E**ngineering **W**orkbench, což ve volném překladu znamená, virtuální laboratoř přístrojové techniky a strojírenství na pracovním stole. Jak už z názvu vyplývá, tak se jedná o sofistikovaný software pro tvorbu měřicího pracoviště pomocí osobního počítače. Výhodou LabVIEW je, že pokud si koupíme měřicí kartu, tak můžeme vytvořit (naprogramovat) téměř jakýkoliv přístroj. Tím se ušetří nemalé prostředky za nákup drahých měřicích přístrojů a osciloskopů.

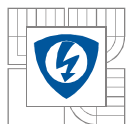
Měřicí karta je měřicí hardware, ke které připojíme měřenou soustavu a propojíme s počítačem. Je to speciální druh A/D převodníku.

Další výhodou je ukládání a analýza naměřených hodnot rovnou v počítači, ale také řízení přes počítač.

LabVIEW je produktem americké firmy National Instruments a ve světě měřicí a řídicí techniky platí za standard, s kterým jsou porovnávány ostatní měřicí programy. LabVIEW je postaveno na systému tzv. virtuálních přístrojů, dále jen VI (Virtual Instrument), které jsou na rozdíl od skutečných přístrojů omezeny pouze výkonem použitého počítače a k němu připojeného hardwaru.

Výrobce LabVIEW tvrdí, že jako jediné grafické programovací prostředí na trhu obsahuje kompilátor generující optimalizovaný kód, který má rychlost vykonání srovnatelnou s rychlostí kompilovaných programů v jazyce C. Aplikace vytvořené v LabVIEW jsou plně srovnatelné s aplikacemi vytvořenými i nízko úrovněnými jazyky jako je C, avšak komfort jejich vytváření je podstatně vyšší. Programátor se zbavuje starostí s řadou syntaktických detailů konvenčního programování.

Uživatelské grafické prostředí je tvořeno dvěma základními okny a to ovládacím panelem (Front Panel) a blokovým schématem programu (Block Diagrams). Blokové schéma je ekvivalentní se zdrojovým kódem textu v klasických programovacích jazycích (C, Pascal). Dále blokový diagram obsahuje nástroje pro vstup a výstup, výpočty, řízení, matematické funkce. Ty jsou zobrazeny jednotlivými ikonami (uzly-nodes), které jsou navzájem propojeny (dráty-wires). Takto vytvořený program je na rozdíl od jiných programových jazyků řízen datovým tokem. Ovládací panel je vlastně vzhled měřicího přístroje, kde může obsluha, nebo jiné VI, měnit vstupní hodnoty a řízení.



Zmíněné netradiční pojetí programování je spíše podobné postupu návrhu přístrojů na hardwarové úrovni nebo výkladu jejich funkce než postupu tvorby programových prostředků. Bude totiž více vyhovovat cílové skupině uživatelů, kteří se zabývají návrhem technických prostředků.

Jak již bylo zmíněno výše, LabVIEW je grafickým programovacím prostředím, tvorba programů, virtuálních přístrojů je formou uživatelských nabídek, které mají jak grafickou podobu (ikony), tak mohou mít i textovou. Při programování se využívá bohatého množství nabídek a různých palet s přednastavenými VI a funkcemi (Fourier, Bessel, Runge Kutta, atd.), tak i logických bloků. Při programování se myš stává nezbytností. Od původního záměru intuitivního programování se LabVIEW už poněkud vzdálil a pro jeho plnohodnotné využití je ho třeba studovat stejně jako jiné programovací jazyky.

Pro spuštění již odladěného programu v jazyce G (programovací jazyk programu), je nutné mít na spuštěném počítači nainstalovaný LabVIEW, a nebo jeho Run-Time Modul ve stejné verzi nebo vyšší, ve které byl program vytvořen.

Aplikace vytvořená v LabVIEW není samostatně spustitelná bez programového prostředí LabVIEW, aby byla zajištěna vysoká přenositelnost aplikace a mohla být volně šířitelná, vytvořila firma National Instruments neplacený doplněk Run Time Modul, který emuluje prostředí LabVIEW. Podmínkou pro běh aplikace v Run Time Modulu je překonvertování VI aplikace na samospustitelný soubor s příponou EXE.

Virtuální přístroje, aplikace vytvořené v LabVIEW, mají příponu souborů VI. Ty jsou v binárním proprietárním tvaru, formát se liší pro různé verze. Program s příponou \*.vi vytvořený ve starší verzi, nemusí být spustitelný v novější. Ale zpětná kompatibility mezi verzemi je zaručena.

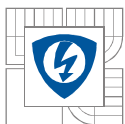
Obsáhlejší aplikace většinou vyžadují ke spuštění více VI, proto je možno potřebnou sadu VI uložit do knihoven LLB s vyznačením nejvyšší úrovně. V knihovnách jsou VI komprimovány tak, že zabírají méně místa na disku.

## 4.1 Programování v LabVIEW

Rozdíl mezi programováním v jazyce G a jiných vývojových jazycích je v řízení programu tokenem dat, nikoli tokenem příkazů. Chceme-li určit posloupnost provádění operací, musíme použít zvláštních konstrukcí (objektů). Pokud tak neučiníme, jsou operace aktivovány tehdy, mají-li na všech svých vstupech data. Je-li takových možností více, určí sled kompilátor.

Zápis algoritmu v LabVIEW se děje propojením uzlů (nodes) pomocí spojů (wires). Spoje odpovídají proměnným v jiných jazycích. Spoje začínají v tzv. ukončujících bodech (terminals). To je vazba na objekty panelu. Uzly jsou výkonné části a jejich základní sada je dodávána se systémem. Mezi ně patří funkce (functions) – ty jsou needitovatelné, jiné editovatelné jsou subVI.

LabVIEW podporuje modulární programování. Složitější program VI může být složen z nižší úrovně objektů subVI, což je obdoba procedur. Parametry do VI jsou předány z vyšší úrovně pomocí konektoru (obdoba seznamu parametrů) do grafického ovládacího panelu. Do VI nejvyšší úrovně zadává parametry uživatel pomocí klávesnice a myši, výstupy jsou zobrazovány zvoleným způsobem (zobrazeny v grafech, uloženy do souboru, tištěny, atd.). Výhodou tohoto postupu je, že každý VI lze jednoduše odzkoušet a potom z ověřených VI skládat složitější.



Pro tvorbu VI jsou k dispozici tzv. funkce, které jsou součástí systému LabVIEW. Funkce lze přirovnat k operátorům a příkazům konvenčních jazyků. Počet funkcí není omezený a může být rozšířen i uživatelsky rozšířen o nové funkce. Tyto nově vytvořené funkce LabVIEW zahrne do grafických menu pro tvorbu VI v jazyce G. Takto jsou tvořeny a dodávány některé utility a prostředky pro zpracování signálů či práci se soubory. Výhodou VI bloků je, že můžeme z demonstračních příkladů nebo jiných již kompletních programů vykopírovat námi potřebný blok či celou sestavu a použít v jiném programu.

Řada funkcí je tzv. polymorfní, to znamená, že funkce sčítání může sčítat jak skalární proměnné (např. typu double), ale také pole libovolných dimenzí i komplexní čísla, a to bez potřeby nějaké volby. LabVIEW dovoluje i tvorbu polymorfních subVI.

Dále také existují datové typy booleovské, znakové a speciální. Každý z těchto typů je vyjádřen v diagramu jinou barvou.

Možnost přiřadit a kontrolovat fyzikální rozměr, zda výsledný objekt tomuto rozměru odpovídá. Pro manipulaci s fyzikálními rozměry je k dispozici několik funkcí. Použití rozměrů sice komplikuje tvorbu programu, ale zase máme k dispozici určitý kontrolní mechanismus, který ověřuje naprogramovaný algoritmus.

LabVIEW podporuje možnost multiprogramování. Je možné vytvořit i několik VI, které pracují samostatně, není mezi nimi žádná datová vazba představující spoje v diagramu. Jsou zde i známé prostředky asynchronního řízení např. semafore, fronty, výskyty (occurrence). Je možné mít spuštěno i více VI a editovat je.

Možnosti spouštění a počet simulací omezuje pouze výkon počítače. LabVIEW dokáže také spouštět jiné aplikace a komunikovat s nimi pomocí speciálních bloků. Pro elektrotechniky je zvláště zajímavá podpora s programem MATLAB.

## 4.2 Komunikace s HW

K měření se používají speciální měřicí karty, které mohou být interního (jako zásuvné moduly PCI, PCMI), nebo externího provedení (připojitelné přes USB). Měřicích karet je velké množství (od jednoduchých převodníků až po signálový procesor), každý druh karty je vhodný k jinému druhu měření. Přímo jsou podporovány jen firemní karty National Instruments a pro tyto karty je k dispozici rozsáhlá knihovna DAQ. Karty dokážou paralelní sběr dat i jejich zpracování, omezující je pouze výkon počítače.





Obr. 4.1 A/D převodník, měřicí karta NI USB-6251

#### 4.2.1 Parametry měřicí karty USB-6251

- 16 analogových vstupů (16-bit)
- 1.25 MS/s na jeden kanál (1 MS/s agregace)
- 2 analogové výstupy (16-bit, 2.8 MS/s)
- 24 digitálních I/O (8 clocked)
- 32-bit čítač

Dále jsou k dispozici tzv. přístrojové ovladače (instrument drivers), které dovolují využívat v LabVIEW již konkrétní přístroje, pro atypické či neznámé přístroje si je možné takový ovladač v LabVIEW naprogramovat. Pro komunikaci mezi měřicími přístroji, zdroji a LabVIEW se používá komunikačních standardů: USB, GPIB (IEEE 488, IMS-2), RS232C a po doplnění i VXI.

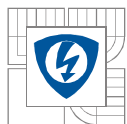
#### 4.2.2 Zpracování dat

LabVIEW má implementovanou knihovnu pro zpracování naměřených či vygenerovaných dat, nazývá se Analysis. Obsahuje dostatečně velké množství VI, které pokrývají skoro veškeré požadavky na zpracování signálů.

Jsou zde VI pro zpracování v oblasti kmitočtové i časové, návrh a realizace číslicových filtrů, rozsáhlé statistické VI, ale i interpolace, regrese, operace s vektory a maticemi, atd.. Celkem je zde přes 200 VI bloků, ale počet VI závisí na druhu licence, kterou jsme si zakoupili. K vyzkoušení a ukázce je zpracována řada demonstračních VI. Z důvodu složitosti některých VI je potřebná dobrá znalost teorie signálů.

V prostředí LabVIEW je implementována řada matematických metod, např. pro řešení diferenciálních rovnic, Fourierových transformací, atd.





### 4.2.3 Požadavky OS

*LabVIEW ve verzi 8.5 je kompatibilní s OS:*

- Microsoft Windows 2000/XP
- Mac OS X Mac OS X 10.3 nebo novější
- Linux Kernel 2.2.x, 2.4.x, nebo 2.6.x

## 4.3 Ukládání do souborů

LabVIEW je vybaveno řadou funkcí pro prezentaci a ukládání dat. Základní přístupy jsou zobrazení na monitoru v rámci aplikace VI, uložení do ASCII souboru a do binárního souboru. Předností ASCII souboru je jeho čitelnost a relativní univerzálnost, avšak ukládání je pomalé a zdržuje aplikaci stejně jako ukládání do strukturovaných dokumentů jednotlivých výrobců, i když se jedná o binární soubory. Nejrychlejším způsobem uložení je binární soubor VI (formát LabVIEW), uložení do takového souboru může být až mnohonásobně rychlejší než do jiných dokumentů.

### Spolupráce s programy fy. Microsoft

LabVIEW na platformě Windows je klientem ActiveX s podporou API, což mu umožňuje rozšířenou spolupráci v rámci tohoto operačního systému. Jak již bylo zmíněno dříve, aplikace VI v prostředí Windows může zavolat další obslužný program a předat mu naměřené hodnoty, aby si je uživatel mohl sám zpracovat. Případně je uložit v příslušném formátu. V současné době výrobce uvádí, že na této platformě jsou plně podporované formáty Microsoft: PowerPoint, Mail, Access, FoxPro, Project, SchedulePlus, Word, Excel. Programátor disponuje sadou funkcí, kterou je schopen vytvořit požadovaný dokument včetně vyšších struktur, nebo pouze naměřená data přidat do souboru vytvořeného těmito prostředky.

### Obecně podporované formáty

Pro distribuci dat naměřených aplikací VI je možno využít některý předpřipravený modul pro zjednodušené generování dokumentů ve strukturovaných dokumentech. Jedná se především o formáty RTF, TeX a jeho odnože, dále pak formáty užívané pro publikování na internetu, HTML a jeho odnože včetně skriptů, PHP, atd.

### Přenos dat po síti

Jak již bylo zmíněno dříve v LabVIEW jsou integrovány prostředky pro přenos dat po síti. Nejedná se jen o přenos dat v rámci měřicí aparatury nebo lokální sítě, ale i výstup do vzdálenějších sítí, např. formou FTP a URL přenosů, případně je možné vytvoření vlastního serveru nebo mail klienta.

### Externí zobrazovače

Pro běžné výstupy jako je televize, monitor, řádkové LCD, průmyslový zobrazovací panel atd., jsou v LabVIEW připraveny moduly.

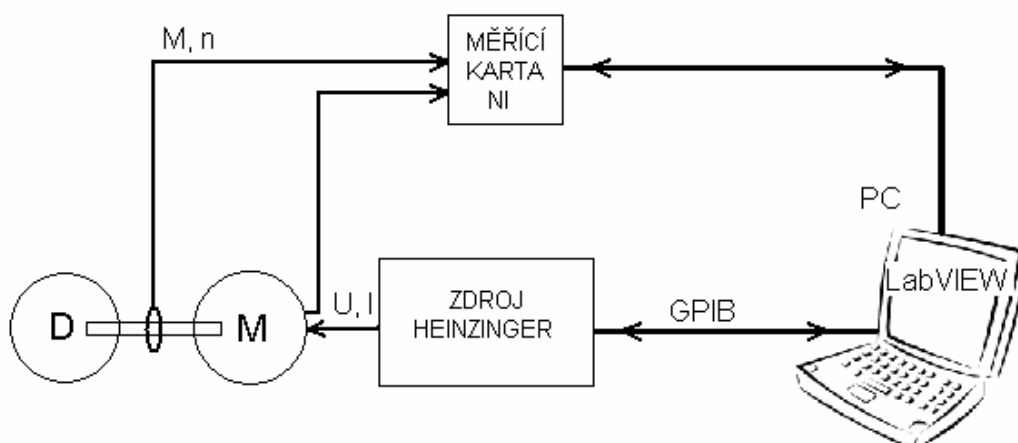
S pomocí těchto modulů lze vytvořit celkem jednoduše nenáročný, ale přehledný zobrazení procesu pro obsluhu.

LabVIEW je též vybaveno moduly pro přímý tisk v prostředí Windows, nejčastěji pomocí API funkcí přes nainstalované ovladače tiskárny. V případě nepřímo dostupných tiskáren je pak doporučována spíše forma PostScriptu.

## 5 NÁVRH A REALIZACE VIRTUÁLNÍ LABORATOŘE

V tomto bodu se budeme zabývat vhodným návrhem virtuálního přístroje pro automatizované měření stejnosměrného motoru se sériovým buzením. Dálkové měření bude realizováno pomocí programu LabVIEW ve verzi 8.5.

Prvně jsme provedli rozbor úlohy, ve kterém jsme si definovali parametry, které budeme chtít měřit a vyhodnocovat. Měřené veličiny jsou: proud  $I$ , napětí  $U$ , moment  $M$ , otáčky  $n$ . Vyhodnocovat budeme výkon  $P$  a budeme chtít zobrazit momentovou charakteristiku do grafu.



Obr. 5.1 Blokové schéma měřicího systému pro ss motor

### Měřicí soustava obsahuje:

- stejnosměrný motor se sériovým buzením
- dynamometr
- optický snímač otáček
- tenzometr
- zdroj napětí Heinzinger
- počítač s programem LabVIEW
- měřicí kartu National Instruments

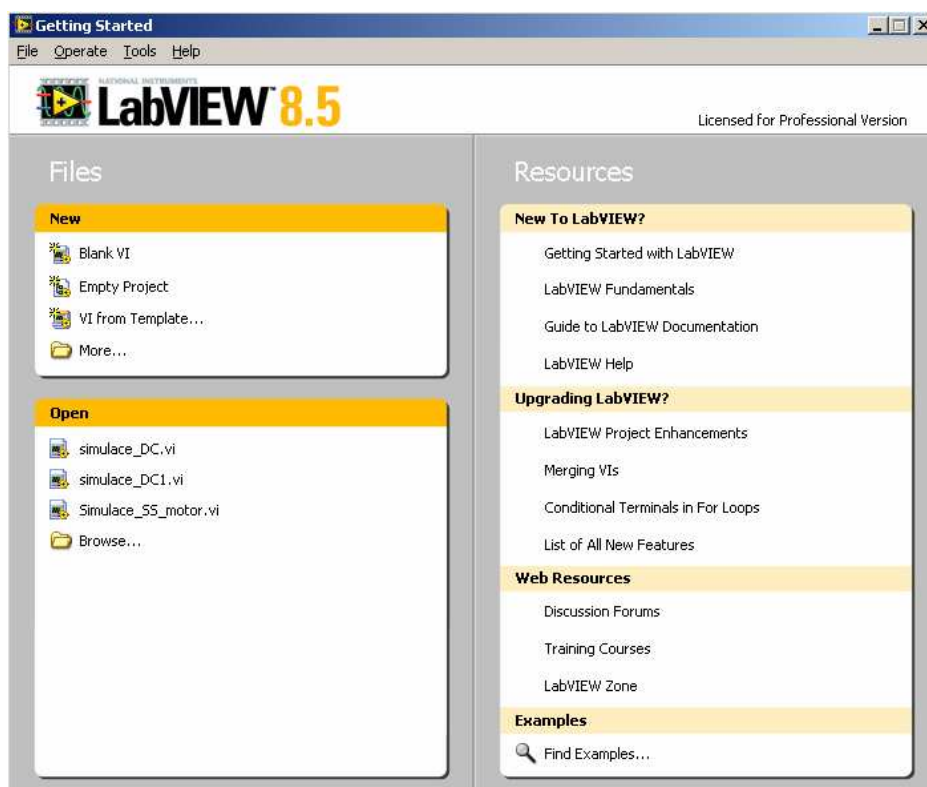
Dynamometr je přes hřídel spojen s měřeným motorem, který je napájen z regulovatelného zdroje napětí Heinzinger. Na vnější plášť dynamometru je připojen tenzometr, který měří zátěžný moment stroje. Výstupní signál z tenzometru je připojen na převodník, který transformuje signál na napětí, které je přivedeno do měřicí karty a ve VI zpracováno a uloženo jako moment  $M$ . Měřicí karta typ USB 62-51, kterou máme k dispozici, dokáže snímat pouze veličiny napětí.

Ze svorkovnice motoru je vyvedeno výstupní napětí  $U_{vyst}$  a výstupní proud  $I_{vyst}$ . Na hřídeli motoru jsou snímány otáčky  $n$ . Veškeré výstupní signály, které chceme zpracovat, musí být převedeny na napětí a připojeny na měřicí kartu. Měřicí karta je napojena přes USB port k osobnímu počítači, kde jsou naměřené signály analyzovány pomocí programu LabVIEW.

Na napájecím zdroji Heinzinger nastavujeme napětí a v důsledku výše popsaných principů se motor roztočí.

## 5.1 Tvorba virtuálního přístroje

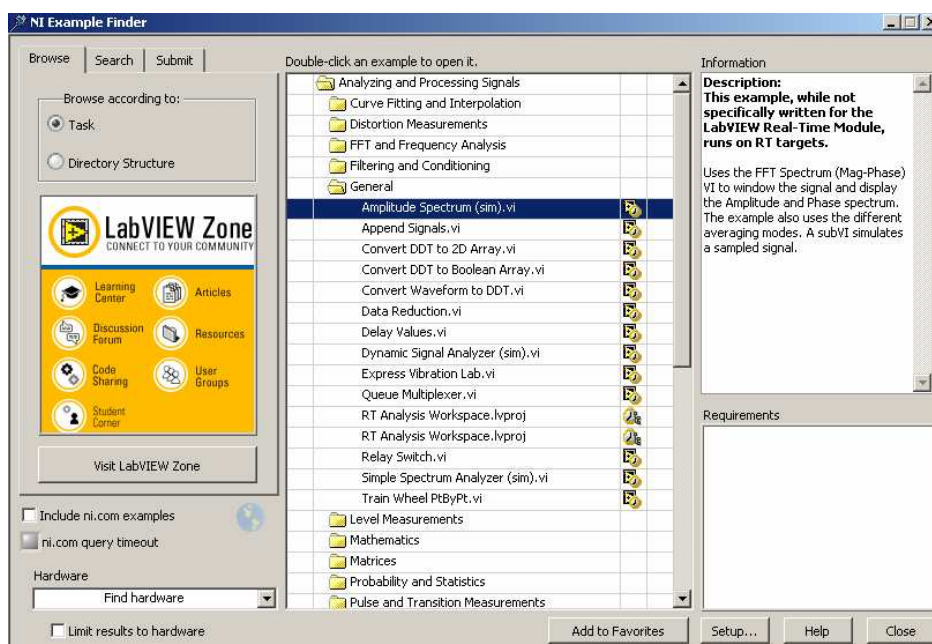
Po otevření programu LabVIEW se ukáže úvodní obrazovka, kde v levé horní tabulce (New) je výběr pro vytvoření nového VI. V dolní tabulce jsou odkazy na otevření poslední použité VI. Pravá tabulka je orientovaná informativně o nových možnostech programu, webové podpoře.



Obr. 5.1.1 Úvodní obrazovka pro tvorbu projektu

Užitečným odkazem je VI from Template, pod kterým jsou uloženy již vypracované šablony nejčastěji užívaných prvků, použití šablon ušetří programátorovi čas i práci.

Další užitečnou funkcí je možnost vyhledání příkladů (Find Examples), po kliknutí na odkaz se zobrazí okno viz. Obr. 8., ve kterém je možno vyhledávat a editovat jednotlivé VI šablony sestav. Rozdíl mezi Find Examples a VI from Template je, že u Find Examples jsou uvedeny kompletní řešené VI aplikace, ale u VI from Template jsou pouze „startovní“ šablony.

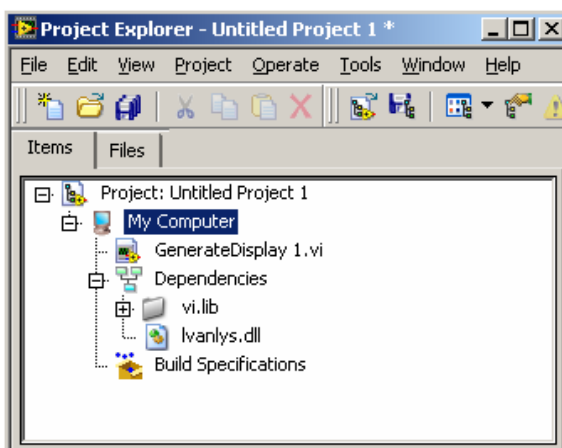


Obr. 5.1.2 Okno vyhledávání a správu příkladů

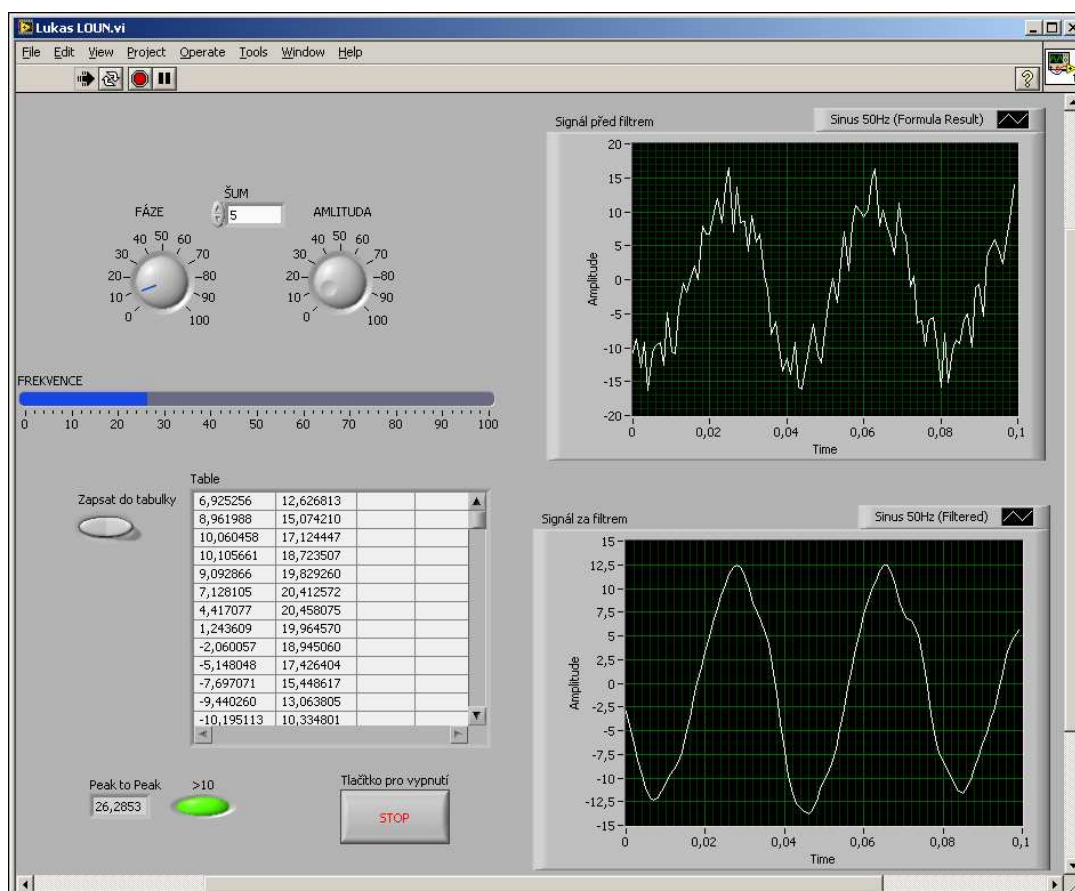
Pro vytvoření nového projektu klikneme na Empty project a zobrazí se okno Project explorer. Ten slouží ke spravování projektů, lze v něm prohlížet a editovat hierarchii VI. Pokud tvoříme složitější soustavu složenou z několika VI, kdy jedno VI je složeno z několika dalších subVI je pro nás práce v projektu nutností.

Výhoda je kompletní správa projektů, pokud je otevřeno paralelně více oken, lze přes Project Explorer uložit všechny změny týkající se jednoho projektu najednou. Soubory vytvořené v Project Explorer mají příponu LVPROJ.

Kliknutím na File/ New VI se otevrou základní okna pro vytváření programů (virtuálních přístrojů), Front panel a Block Diagram.



Obr. 5.1.3 Okno Project Explorer



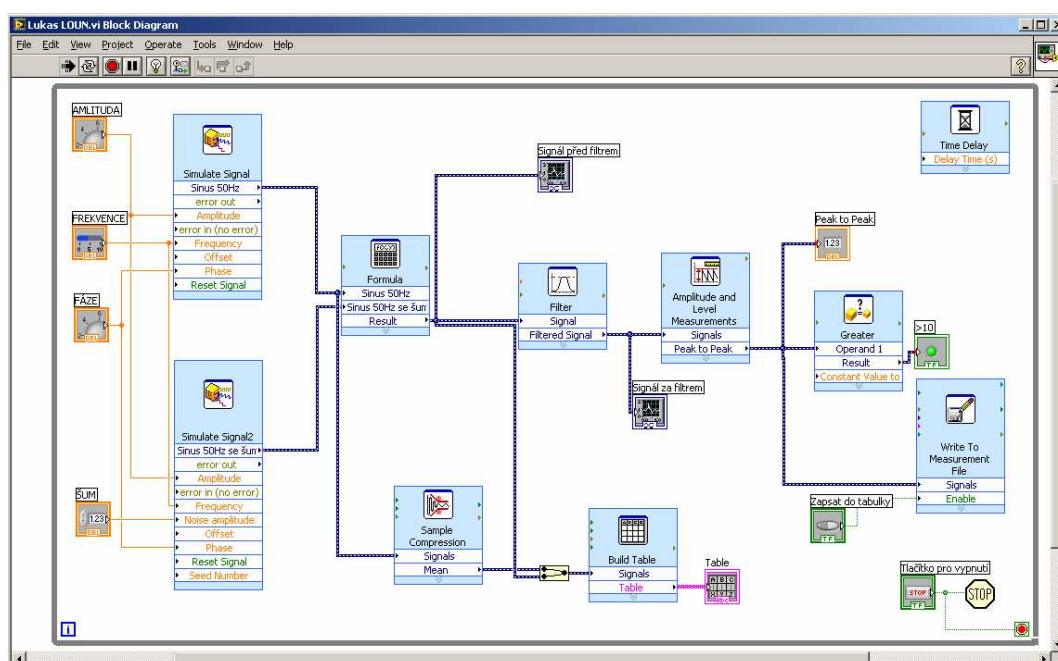
Obr. 5.1.4 Front Panel – vnější vzhled přístroje

Ve Front Panel se vytváří vnější vzhled přístroje, rozmístění prvků. Jedná se vlastně o grafický výstup virtuálního přístroje.

Tento panel se používá za chodu VI, zde se nastavují vstupní veličiny, ale také se zde ovládají přepínače, zapisovače, atd. Front Panel, jak už naznačuje anglický název, nahrazuje přední panel reálného měřicího přístroje, ať už je to osciloskop, multimetr či jiné elektronické zařízení.

Front Panel zobrazuje pouze vstupní a výstupní ovladače a zobrazovače. Myšlenka National Instruments je taková, že přední panel stejně jako u skutečného přístroje je přístupný i zaškolenému laikovi, který přístroj dokáže ovládat pomocí ovládacího panelu přístroje. Ale útroby přístroje, které udávají charakter přístroje a způsob použití musí navrhnout a naprogramovat odborník, který dokonale rozumí dané tematice. V LabVIEW jsou tyto útroby přístroje představovány bloky v Block Diagram. Programátor navrhne na zakázku program (VI) v Block Diagram, rozmístí ovládací prvky ve Front Panel a upraví grafiku. Potom koncový uživatel může používat daný virtuální přístroj, aniž by rozuměl vnitřnímu uspořádání v Block Diagram. K úspěšnému měření stačí pouze ovládat Front Panel, tak jako by to byl skutečný měřicí přístroj.





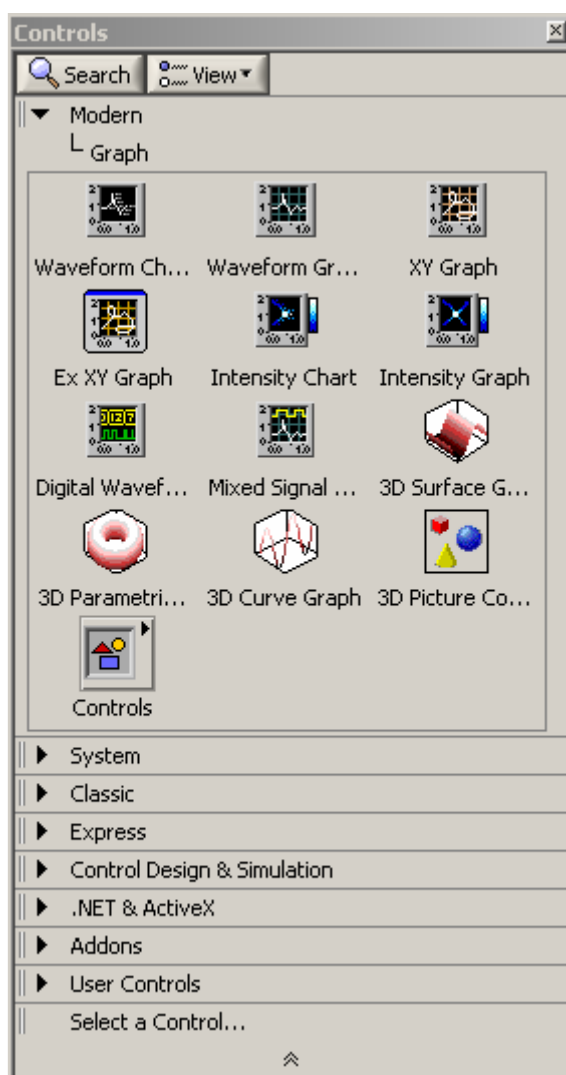
Obr. 5.1.5 Block Diagram

**Schéma algoritmu se vytváří z několika typů entit:**

- **terminály** zajišťují komunikaci s předním panelem
- **funkce** výkonný aparát pro zpracování dat
- **interface** umožňuje přímý přístup k podporovanému hardwaru
- **ovladače** ovládání podporovaných periférií počítače
- **vodiče** zajišťují tok dat mezi ostatními entitami

Vytváření jednoduchých aplikací je díky této struktuře a řadě předpřipravených entit jednoduché a intuitivní. Problémy se vyskytují až při vytváření složitějších aplikací. **LabVIEW obsahuje všechny běžné nástroje pro ladění programů.**

K vytvoření prvku na ovládacím panelu Front panel dojde kliknutím pravým tlačítkem myši do plochy okna. Zobrazí se panel Controls, ve kterém můžeme vybírat z různých vstupních a výstupních objektů. Je zde na výběr z několika druhů otočných regulátorů (knob), posuvníků (slide), tlačítek (button), zobrazovačů (progress bar), numerických indikátorů, ale i waveformů, grafů (2D, 3D), tabulek, matic, atd. Okno Controls je uživatelsky přizpůsobitelné a je možné si v něm navolit nejčastěji používané objekty. Přidání objektu provedeme přetažením myši do okna Front Panel. V tomto okně se nenastavují žádné vazby mezi jednotlivými bloky, to se provádí v Block Diagram, pouze se zde nastavuje vzhled a rozmístění ovládacího panelu.

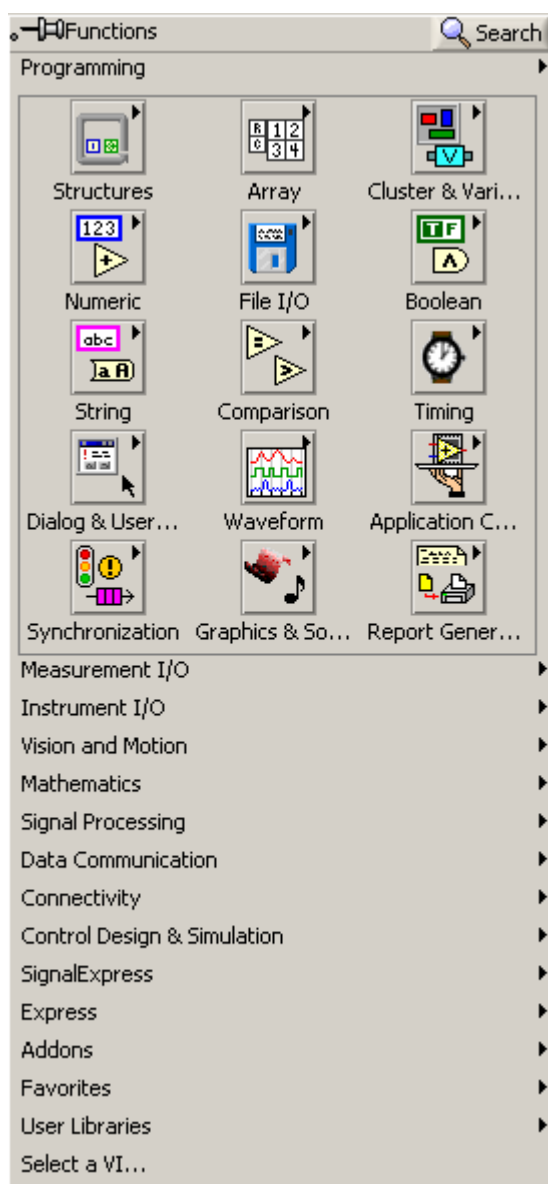


Obr. 5.1.6 Controls panel pro tvorbu ve Front Panelu

Stejně jako Front Panel má pro vytváření bloků okno Controls, tak má i Block Diagram okno Functions. Okno zobrazíme stejně jako u Controls, přes pravé tlačítko myši. Okno si můžeme permanentně zobrazit kliknutím na přepínaček v levém horním rohu.

Paleta Functions je mnohem bohatší než u Controls. Jelikož okno Block Diagram je hlavním prostředím pro tvorbu VI, obsahuje programátorské funkce: numerické funkce (ADD, MULTIPLY, INCREMENT, atd.), booleovské funkce (AND, OR, NOT, atd.), matematické (diferenciální rovnice, Fourierovy transformace, atd.) a spoustu jiných bloků, ale i hotových VI, které mají zjednodušit programování. Jsou zde již přednastaveny bloky pro odesílání reportů emailem, atd.

Není nezbytné zde vyjmenovávat všechny bloky a jejich vlastnosti, protože je jich více než 200, a není to ani náplní této práce. Jenom bych se ještě rád zmínil o skutečnosti, že všechny bloky jsou zpracovány v poměrně rozsáhlém helpu a je zde i pár příkladů použití.

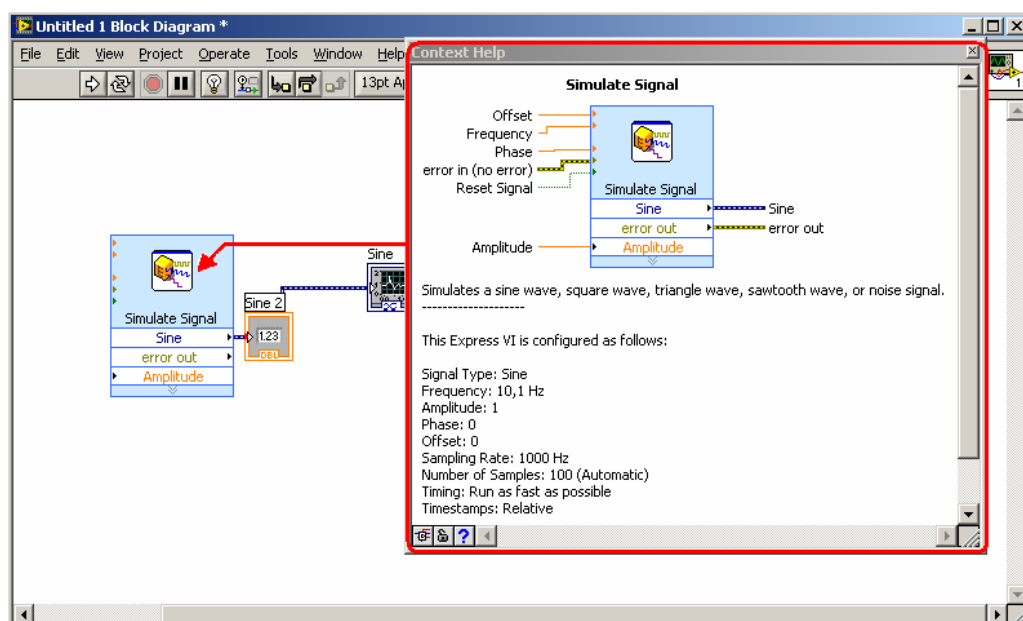


Obr. 5.1.7 Function panel pro tvorbu virtuálních přístrojů v okně Block Diagram

## 5.2 Automatizované měření pro stejnosměrný motor

Vytvoříme si nové VI v LabVIEW a připojíme měřicí kartu přes USB k počítači. V Block Diagramu přes panel Function vybereme pod záložkou Measurement I/O dále DAQmx – Data Acquisition a DAQ Assistant. Výhodné je použít kontextovou nápovědu, kterou vyvoláme kombinací kláves Ctrl + H, otevře se nám okno, ve kterém se po pouhém najetí kurzoru na objekt zobrazí nápověda. Kontextová nápověda je dynamická, přizpůsobuje se aktuálnímu nastavení přístroje a zobrazuje i vnitřní nastavení bloku.



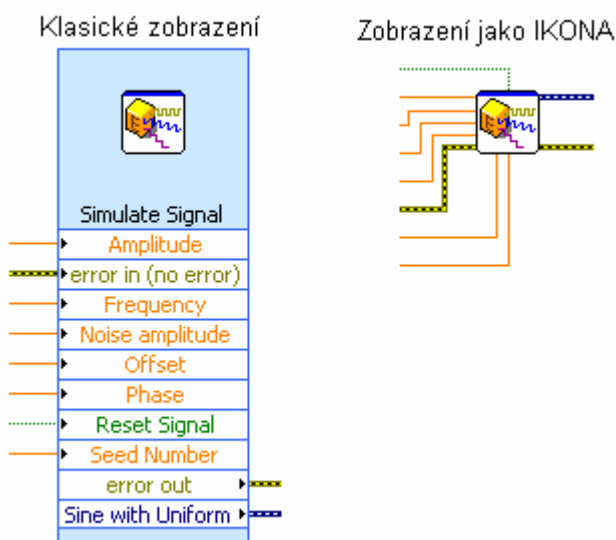


Obr. 5.1.8 Context Help - Červeně je zvýrazněn kurzor myši a okno kontextové nápovědy

Po vložení DAQ Assistance se automaticky otevře okno Create New Express Task, kde vybereme Acquire Signals, dále Analog Input, Voltage a dokončíme tlačítkem Finish.

Objeví se okno DAQ Assistance, ve kterém přiřadíme fyzickým vstupům měřicí karty virtuální výstupy, vybereme si měřicí veličinu (napětí, proud, frekvenci, atd.), zvolíme si vzorkovací frekvenci.

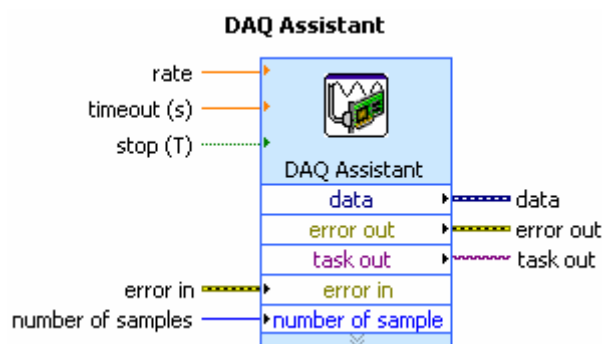
Jednotlivé bloky mohou být v Block Diagramu zobrazeny dvěma způsoby. Buďto v klasickém (plném) zobrazení, nebo jako ikony viz. obr. 5.1.9.



Obr. 5.1.9 Možnosti zobrazení bloků

Pro přehlednost je lepší programovat s klasickým zobrazením a nikoliv s jejich náhledy. V náhledech se hůře orientuje a při hledání problému není evidentní, na který vstup je wire (drát) přiveden. Ale pokud vytváříme více podobných prvků implementovaných do jedné smyčky, je vhodné zvolit zobrazení jako ikona, bloky zabírají menší plochu a snáze se tím orientuje ve schématu. Ideálním řešením je vytvořit propoje mezi bloky v klasickém zobrazení a poté je změnit na zobrazení jako ikona. Tyto bloky posléze rozmístit ve schématu tak, aby co nejvíce vyhovovali funkčně i ergonomicky. Poté přes pravé tlačítko myši klikneme na jednotlivé wire a vybereme možnost Clean Up Wire a program nastaví nejvhodnější propoj mezi jednotlivými bloky.

### 5.2.1 Nastavení DAQ Assistant



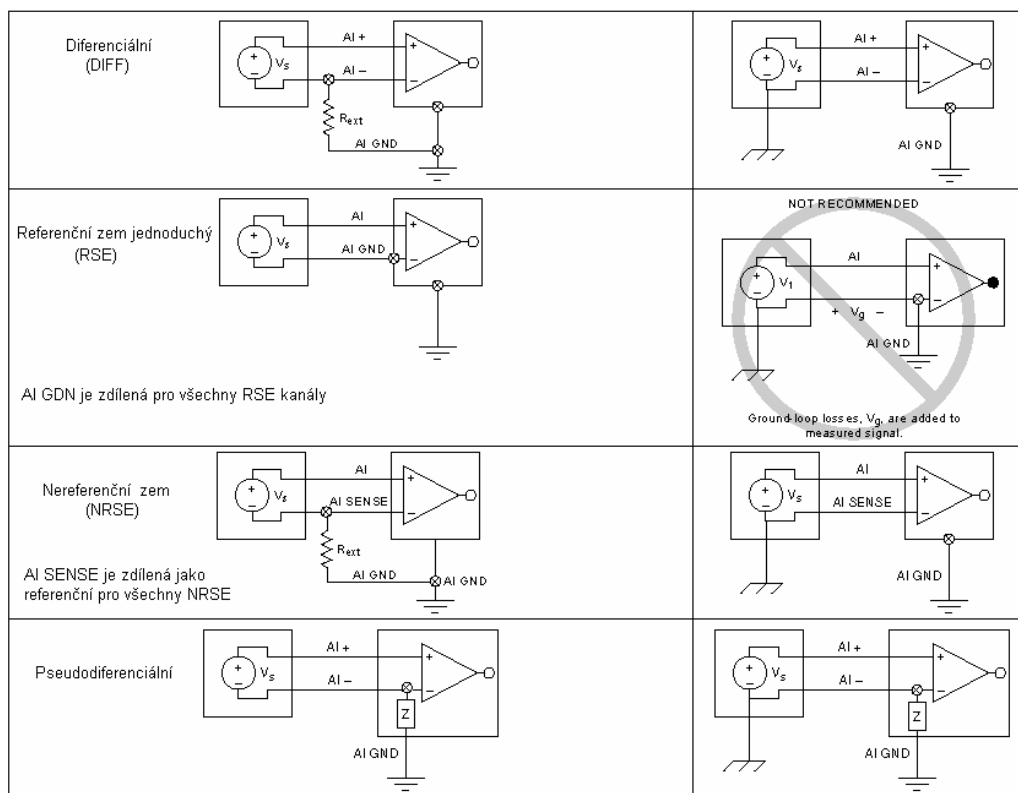
Obr. 5.1.10 DAQ Assistant

DAQ Assistant je základní programový blok představující v Block Diagramu zdroj měřených signálů. V tomto bloku nastavujeme parametry měřicí karty, žádná automatizovaná úloha, která měří reálné veličiny se neobejde bez tohoto VI. Měřicí karta NI USB-6251 umožňuje snímat pouze napětí, a to v rozsahu  $\pm 10V$ , karta má 16 analogových vstupů, může měřit až 16 různých veličin. Vstupy jsou separátní, to znamená, že pro každý tento vstup můžeme v DAQ asistentovi nastavovat jiné parametry. Pro každý vstup můžeme nastavit rozsah pro min. a max. hodnotu napětí, ve které bude měřicí karta pracovat.

Dalším důležitým nastavením je *Terminal Configuration* – pro výběr druhu zemnění. Karta NI USB-6251 má tyto možnosti:

- **Differential** – pro rozdílné impedance zemí, třeba termoelektrické články nebo bateriové zařízení. Musí být připojen k zemi přes externí odpor.
- **RSE** – se společnou zemí
- **NRSE** – stejně jako RSE, ale napětí se může měnit s ohledem na systémovou zem
- **Pseudodifferential** – použijeme, pokud měříme kladné a záporné vstupy na jednom fyzickém kanálu připojené na zem přes rozdílné impedance.

Zvolená konfigurace vždy záleží na druhu použitého hardwaru a měřených přístrojů.



Obr. 5.1.11 Druhy zapojení měřicí karty

$R_{ext}$  je externí odpor, který musíme přidat

V prvním sloupci tabulky je zapojení pro **plovoucí zdroj signálu** (není pevně připojen k zemi), např. termoelektrické články, signály pro úpravu signálů s oddělenými výstupy a bateriové zařízení.

Ve druhém sloupci tabulky jsou reprezentována zapojení **s uzemněným zdrojem signálu**, přístroje s neoddělenými výstupy.

Pro naši úlohu jsme použili diferenciální zapojení, jelikož snímáme veličiny, kde nemůžeme zaručit stejný potenciál země. Používáme optický snímač otáček, klešťový ampérmetr a pro snímání momentu tenzometr s převodníkem.

Signály k měřicí kartě jsou přiváděny metalickými vodiči a připojeny ke svorkovnici karty podle zvoleného druhu zapojení. Je důležité přesně dodržet zapojení na příslušné svorky a polaritu zemí! Zapojení si můžeme zkontrolovat pomocí Connection Diagramu, který nám graficky zobrazí správné schéma zapojení s barevným označením polarit vodičů.

V DAQ asistentovi si můžeme pojmenovat měřené signály, např. *Napětí kotvy*, *Proud kotvy*, *Moment*, *Otáčky*, usnadníme si tím pozdější programování a také ulehčíme hledání a následnou analýzu chyb jak už softwarového či hardwarového typu.

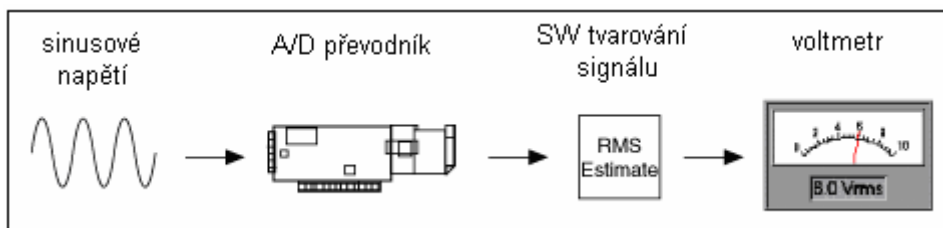
Důležitým nastavením měřicí karty je její samplovací frekvence, ta značí kolik vzorků má načítat při jaké frekvenci. Toto nastavení nám určuje vnímavost celé soustavy, nastavení je stejné pro všechny měřené veličiny. Čím vyšší hodnotu smplování a frekvence nastavíme, tím získáme více hodnot za daný časový okamžik. Je ale nutné volit samplovací frekvenci v rozumné míře. Bude nejspíš zbytečné volit 1MS/s pro měření synchronního motoru v ustáleném stavu, který budeme měřit přibližně 5 minut. Jelikož datový tok bitrate bit/s by zbytečně zatěžoval měřicí systém a s rostoucím bitratem by i úměrně rostl objem dat ukládaný na médium (HDD, USB, SD, atd.). Naopak tomu bude u rychlých dějů jako je například měření vypínacích charakteristik jističů, které trvají řádově milisekundy, tam budeme volit co největší počet samplů, jelikož děj trvá velmi krátkou dobu.

DAQ asistent má funkci Triggering, tu můžeme použít pro synchronizování spuštění měření s nějakým dalším přístrojem např. u výše zmíněného měření vypínací charakteristiky.

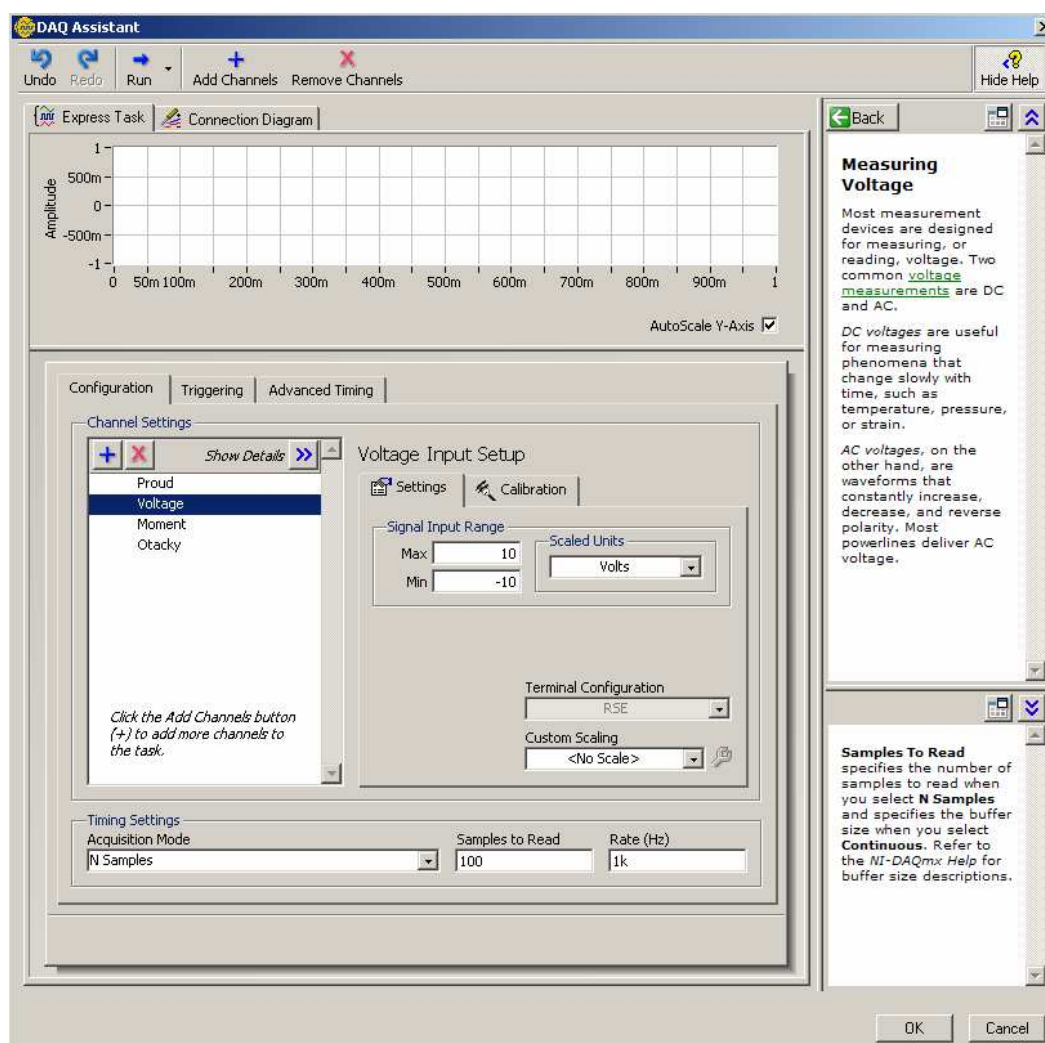
Poslední důležitá vlastnost, která stojí za zmínku, je kalibrace přístroje. Ta se provádí stejně jako u jiných měřicích přístrojů (referenční hodnota a kalibruje se přístrojem minimálně o dva řády přesnějším).

V našem případě jsme nastavili DAQ následovně, pro každou měřenou veličinu jsme si přidali jeden kanál tlačítkem Add Channels a vybrali jsme volbu napětíový vstup. Podle manuálu přístroje jsme připojili vodiče z měřeného motoru na svorkovnici měřicí karty. Změny potvrdíme tlačítkem OK.

Nyní máme DAQ převodník nastavený na snímání veličin proudu  $I$ , napětí  $U$ , momentu  $M$  a otáček  $n$ .



Obr. 5.1.12 Blokové schéma měření napětí



Obr. 5.1.13 Nastavení vstupů na měřicí kartě National Instruments USB-6251

## 5.2.2 Programovatelný zdroj Heinzinger

Stejnoseměrný zdroj Heinzinger je určen pro napájení zařízení do maximálního napětí 24V. Je schopen dodávat do zátěže maximální proud až 300A. Tyto výstupní parametry pokrývají širokou oblast elektrických strojů používaných v řídicí a automobilové elektrotechnice. Velkou výhodou tohoto stejnosměrného zdroje je maximální výstupní proud, který zaručuje velkou tvrdost zdroje i pro přechodné děje. Pro co největší přiblížení chování stejnosměrného Heinzingeru reálnému zdroji napájecímu pohon (např. akumulátoru) je vybaven možností nastavení vnitřního odporu. Další důležitou vlastností je možnost volby maximálního výstupního proudu, což je podstatné u měření menších strojů, kdy se při ověřování nově konstruovaného stroje mohou projevit nepředpokládané chyby, které by mohly způsobit nadměrný odběr proudu a zničení elektrického stroje.

Všechny uvedené funkce lze nastavit na ovládacím panelu zdroje nebo pomocí komunikačního protokolu pro automatické řízení. Jedná se o rozhraní typu RS-232C a dále o rozhraní typu GPIB (IEEE 488.2). V našem případě využíváme pro komunikaci se zdrojem rozhraní GPIB, které dovoluje použití vyšší přenosové

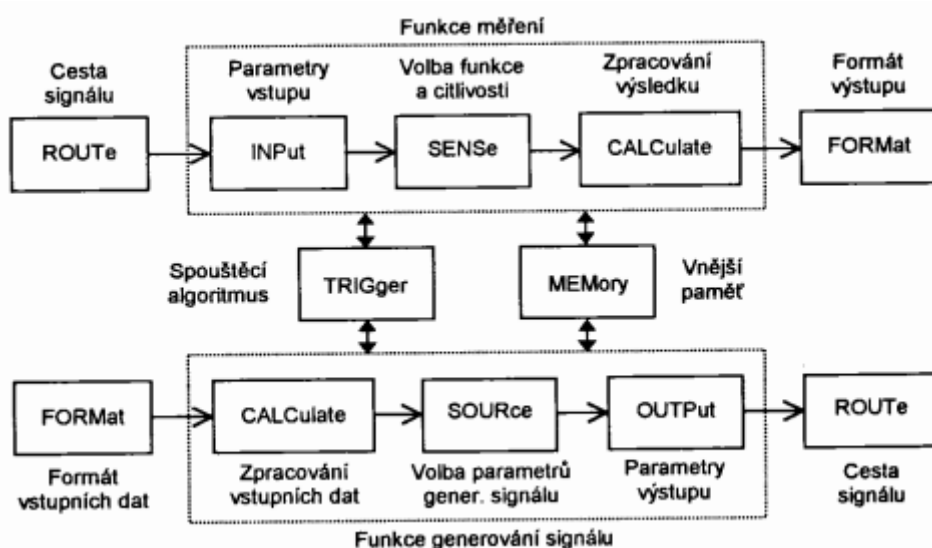
rychlosti a také připojení více zařízení na společnou sběrnici. Zdroj lze ovládat pomocí příkazů jazyka SCPI.

### 5.2.2.1 SCPI (Standard Commands for Programmable Instruments)

Standard SCPI je souhrn příkazů a pravidel pro komunikaci mezi řídicí jednotkou a přístrojem v automatizovaném měřicím systému, nezávislý na technickém řešení ani na technickém protokolu přenosu dat. Přehled příkazů SCPI je uveden vždy v dokumentaci k přístroji včetně ukázek programování.

SCPI definuje:

- hierarchicky uspořádanou množinu SCPI příkazů pro ovládání měřicích přístrojů
- množinu obecných povinných příkazů – common commands, určených k identifikaci stavu a nastavení vnitřních registrů
- formáty dat



Obr. 5.1.14 Přístrojový model SCPI

Příklady povinných obecných příkazů

- **\*RST** (Reset Command) příkaz pro „nulování“ přístroje (ukončení probíhajících operací a nastavení přístroje do klidového stavu), obsahy stavových registrů zůstanou nezměněny
- **\*VOLT\_#** tento příkaz provede nastavení nového napětí zdroje, které je uvedeno v příkazu.
- **\*CURR\_#** tento příkaz provede nastavení omezení výstupního proudu na hodnotu definovanou příkazem. Zdroj se potom chová podle výstupního proudu jako zdroj napětí při výstupních proudech nižších jak hranice a jako zdroj proudu při odběru rovném nastavené hranici.
- **\*CLS** (Clear Status Command) příkaz pro nulování stavových registrů

- **\*IDN?** (Identification Query) dotaz na specifikaci přístroje

### NI 488.2M software pro komunikaci

Je software firmy National Instruments představující průmyslový standard

NI 488.2M Driver a Utility:

- NI 488.2M, 32-bitový driver (gpib\_x.vxd) instalovaný do Windows
- gpib-32.dll dynamicky linkovatelná knihovna
- Win32 Interaktivní řízení (Interactive Control), interaktivní komunikace s GPIB zařízeními s použitím NI 488.2 funkcí
- GPIB Spy – monitorovací program
- GPIB konfigurační utility
- diagnostika
- od NI 488.2 verze 1.6 je instalován i Measurement & Automation Explorer

Aby byla umožněna co největší univerzálnost automatizovaného měření, použili jsme pro komunikaci mezi PC a Heinzingrem USB rozhraní, oproti řízení přes sériový port je to dražší varianta, ale umožňuje použití i na novějších notebookcích, které nemají sériový port COM. Pro naši úlohu jsme použili USB/GPIB Převodník od fy. Agilent typ Agilent 82357B USB/GPIB Interface.



*Obr. 5.1.15 USB/GPIB převodník Agilent*

Podporované standardy:

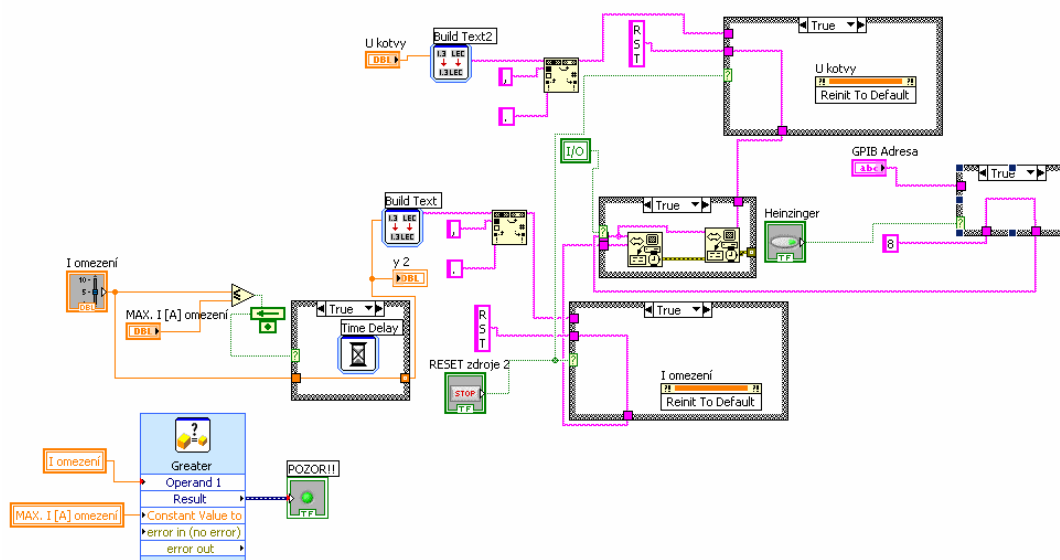
- IEEE-488.1 a IEEE-488.2 compatible
- SICL a VISA 2. GPIB
- USB 1.0 a 2.0



Další parametry převodníku:

- GPIB transfer rate 1,15MB, není potřeba externí napájení (napájeno přes USB +5V).
- Více o parametrech převodníku viz. literatura [7]

Výrobce uvádí, že zařízení podporuje standard Plug and Play, tudíž pro jeho funkčnost není třeba instalovat další software. V našem případě sice Microsoft® Windows XP Professional v sestavení SP2 2600 zařízení rozpoznal, ale dále převodník nekomunikoval se stejnosměrným zdrojem napětí Heinzinger. Vzájemnou komunikaci se podařilo navázat až po použití dalšího softwaru od firmy Agilent, Agilent IO Control, kde jsme museli vypnout autodetekci připojeného přístroje. Agilent má vytvořené knihovny přístrojů, popřípadě možnost downloadu dalších knihoven z internetu. Bohužel Heinzinger PCYs 24-300 se již několik let nevyrábí a výrobce k němu neposkytuje žádné ovladače pro komunikaci s novějšími programy jako je například LabVIEW. Komunikace probíhala přes zásuvnou GPIB kartu uvnitř zdroje pomocí vnitřního protokolu SCPI, který pracuje s textovými příkazy viz. výše v textu. Proto jsme museli v LabVIEW vytvořit modul, kterým lze ovládat zdroj přes virtuální laboratoř na dálku bez nutnosti vypisovat SCPI příkazy zdroje Heinzinger. Na obrázku níže je zobrazeno ovládání zdroje, jak jsme jej navrhli pro řízení přes LabVIEW.




Obr. 5.1.16 Schéma řízení zdroje Heinzinger

Řízení je navrženo tak, že na ovládacím panelu máme dva posuvníky, pro nastavení proudového omezení a výstupního napětí ze zdroje. Proudové omezení je po spuštění programu nastaveno na nulu. Aby bylo možno provést měření, musí učitel či jiná kompetentní osoba nastavit maximální hodnotu proudového omezení v záložce UČITEL. Toto opatření je vytvořeno proto, aby bylo eliminováno nebezpečí poškození motoru neodborným ovládáním. V systému je zakomponováno tlačítko centrálního zapnutí/vypnutí, které platí i pro ovládání zdroje napětí.



Pokud není toto tlačítko zapnuto, nebude nastavování napětí ani proudového omezení v činnosti.

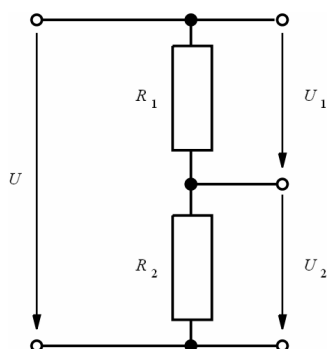
Zdroj napětí Heinzinger je naladěn na GPIB adresu číslo 8, pokud je sepnuto hlavní tlačítko, tak tlačítkem Heinzinger aktivujeme spojení a můžeme nastavovat hodnoty. V případě, že nebude použit jako zdroj napětí Heinzinger PCYs 24-300, ale novější Heinzinger, nebo jiný zdroj napětí pracující na protokolu SCPI s příkazy VOLT\_#, CURR\_#, atd., je pod tlačítkem RESET zdroje dialogové okno s názvem GPIB Adresa, do kterého stačí vepsat číslo GPIB adresy, na kterém daný zdroj komunikuje a stisknutím tlačítka  je nastaveno spojení.

Tlačítko RESET zdroje slouží k vynulování zdroje. POZOR! Pokud přerušíme aplikaci bez předchozího vynulování (resetování) zdroje, zůstane na výstupních svorkách zdroje poslední nastavované napětí.

Měřicí karta NI USB-6251 umí snímat napětí pouze v rozsahu  $\pm 10V$ , pokud chceme měřit s napětím vyšším, musíme připojit napěťový dělič. V našem případě bude maximální měřené napětí 25V, toto omezení je dáno napěťovým zdrojem Heinzinger.

### Napěťový dělič

V elektronice se *dělič napětí* nebo také *napěťový dělič* nebo *odporový dělič* používá pro získání výstupního napětí ( $U_1$  nebo  $U_2$ ), které je úměrné vstupnímu napětí ( $U$ ). Dva rezistory jsou spojeny v sérii, jak je naznačeno na následujícím obrázku. Abychom dosáhli vstupního napětí  $\pm 10V$ , navrhli jsme dělič pracující v poměru 1:3. To znamená, že horní odpor  $R_1$  jsme zvolili  $20k\Omega$  a dolní odpor  $R_2$ , z kterého budeme snímat výstupní signál, jsme zvolili  $10k\Omega$ . Při maximálním napětí 25V na vstupu děliče bude na výstupu 8,33V, což plně postačuje požadavkům na měřicí kartu.



$$U_2 = \frac{R_2}{R_1 + R_2} \cdot U$$

$$U_2 = \frac{10k\Omega}{20k\Omega + 10k\Omega} \cdot 25V$$

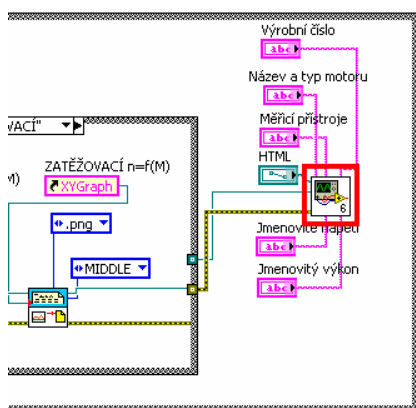
$$U_2 = 8,3\bar{3}V$$

### 5.2.3 Hierarchická stavba – tvoření subVI

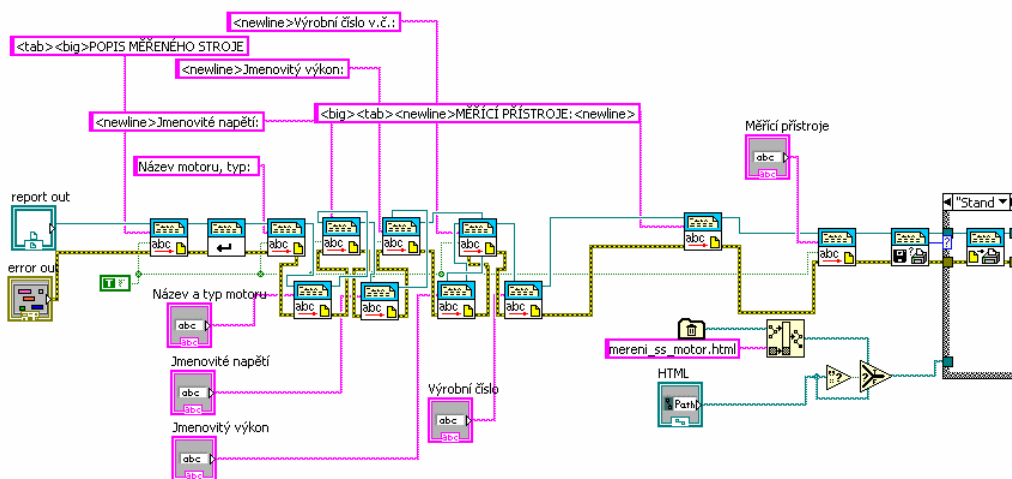
Vzhledem k rozsáhlosti programu jsme programování řešili hierarchicky a některé funkční části jsme vnořili do subVI. SubVI je zvlášť vytvořený program, který je strukturou obsáhlý a může být spuštěn nezávisle na celku. Aby bylo programování čisté a přehledné, vytváří se tyto subVI a zapojují se přes wire do obvodu jako funkční bloky.

Pokud nadřazené VI potřebuje funkci vnořenou do subVI, jednoduše si jej zavolá a proběhne cyklus. Ve spouštěcím VI je vidět celá struktura programu s náhledy na jednotlivé subVI, i když je sice vidět celá struktura, program nebude pracovat správně bez příslušných sub VI.

Náš program obsahuje jedno spouštěcí VI a jedenáct subVI. Díky tomuto uspořádání je program lépe čitelný a snáze se v něm lze zorientovat.



Obr. 5.1.17 Schéma subVI v nadřazeném VI

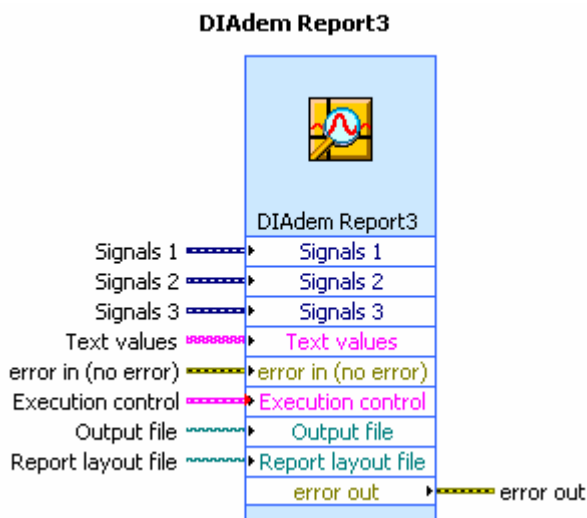


Obr. 5.1.18 Detailní náhled na subVI zobrazené v Obr. 23

Červeným čtverečkem je znázorněno subVI ve spouštěcím VI. Na obrázku níže je zobrazeno stejné subVI po rozkliknutí. Z obrázků je evidentní funkce hierarchického programování, co se týče úspory a přehlednosti.

## 5.2.4 Reportování - možnosti ukládání naměřených veličin a grafů

Naměřené hodnoty uložené v tabulce spolu s vykreslenými grafy můžeme reportovat několika různými způsoby. Nejjednodušším způsobem je blok DIAdem Report.

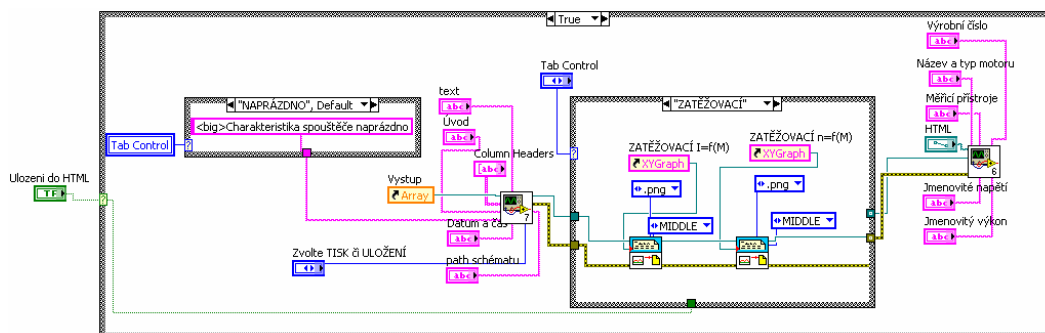


Obr. 5.1.19 DIAdem Report blok

DIAdem Report dokáže exportovat grafy do souborů s příponou *PDF* nebo *HTML*. Výhodou DIAdem Reportu je jednoduchost, k tomuto bloku stačí přivést měřené signály a nastavit ve vlastnostech reportu počet grafů a závislosti jednotlivých charakteristik. Pracuje se v příjemném grafickém prostředí, kde je k dispozici široká škála nastavení vzhledu grafů. Nevýhodou tohoto bloku je, že nedokáže vytvářet kompletní reporty s tabulkami naměřených hodnot a dynamickými textovými poli, které by vkládal přímo uživatel aplikace (ne programátor). Další nevýhodou použití DIAdem Reportu je dlouhé zpracovávání měřeného signálu do grafů, způsobí v LabVIEW celkové „zamrznutí“ celého měřicího aparátu na desítky sekund.

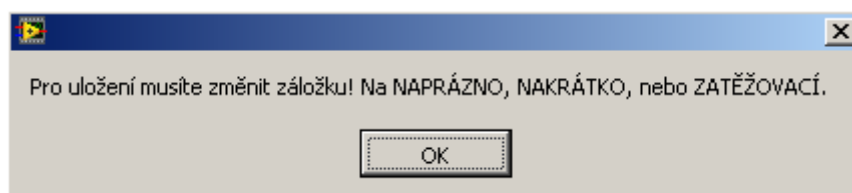
### 5.2.4.1 HTML report

Komplexnější a rychlejší metoda vytváření reportů je pomocí palety Report Generation, kde kombinací různých bloků můžeme vytvořit jakkoliv obsáhlý report. V našem případě se generují tři varianty reportů, a to pro měření naprázdno, nakrátko a zatěžovací charakteristiku.



Obr. 5.1.20 Schéma pro reportování do HTML

V hlavičce generované zprávy se vyplní název měřené charakteristiky, ten je nastaven přes Case Structure na lokální proměnnou, která sleduje zobrazení záložek na Front panelu, tak aby byl název reportu shodný s měřenou charakteristikou. V pravém horním rohu je generován aktuální čas a datum. Tyto hodnoty jsou sbírány z nastavení MS Windows®, proto pro správné generování času zkontrolujte nastavení času a data v operačním systému. Další část reportu je věnována teoretickému úvodu, který je uživatelsky neměnný stejně jako schéma zapojení, protože úloha je koncipovaná pro měření charakteristik motoru – spouštěče. Schéma je načítáno z externího souboru lokální složky pomocí speciálního bloku, nebo může být načteno z webu pomocí URL adresy. Pod schématem se nachází tabulka naměřených a vypočtených hodnot, které jsou sbírány z matice pomocí lokální proměnné do bloku Append Table to Report. Hodnoty z lokální proměnné jsou pouze obrazem matice, do které se zapisují naměřené hodnoty, které nesou informaci pouze o rozměru, ale ne o charakteru veličiny. Proto jsme v Table Reportu nastavili pomocí array názvy jednotlivých sloupců a jejich šířku. Vyexportování grafů do reportu je vyřešeno přes Append Control Image to Report, který sejme aktuální obrazovku daného XY Graphu a vloží ji do souboru. Signál, v našem případě obraz, přivádí k bloku zpracování obrazu VI Server Reference, ten má obdobnou funkci jako lokální proměnná. U získaného obrazu můžeme volit ze dvou formátů pro uložení, a to z JPG a PNG. My jsme zvolili méně známý formát PNG proto, že používá oproti JPG bezztrátovou kompresi, tzn. že daný obraz neztrácí zoomováním na své kvalitě. Bloky pro export grafů jsou implementovány do Case Structure se šesti módy pro přepínání, ovládají se pomocí záložek ve Front Panelu a aktivní může být vždy jen jedna. Šesti proto, že každá záložka ve Front Panelu má přiřazený jeden case v Block Diagramu, který má vnitřní strukturu odlišnou od ostatních case. Vytvořeno je to tak proto, aby se pro každý druh měření vytvořil jeden report s požadovanými charakteristikami a parametry. Výběr charakteristiky pro vtištění se provede ve Front Panelu volbou příslušné záložky. Tisknout lze pouze ze záložek, které zobrazují naměřené charakteristiky, a to jsou NAPRÁZDNO, NAKRÁTKO, ZATĚŽOVACÍ. Ze záložek OSCILOSKOP, VSTUPNÍ DATA a UČITEL tisknout reporty nelze! Při pokusu vtištění či uložení průběhů z těchto dvou záložek se objeví varovná zpráva viz. obr. 5.1.21.

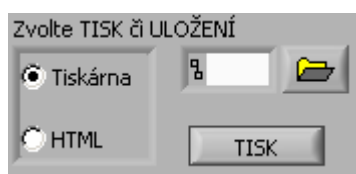


Obr. 5.1.21 Varovné okno pro změnu záložky

Měření nebude pokračovat dál, dokud neklikneme na tlačítko OK.

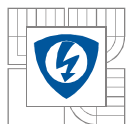
Součástí reportu je i záznam o měřeném zařízení, který si vyplní sám uživatel v záložce VSTUPNÍ DATA. Vstupní formuláře jsou vytvořeny na bázi string vstupů, mezi jednotlivými okny se lze přepínat pomocí tabulátoru v pořadí Název a typ motoru, Jmenovité napětí, Jmenovitý výkon a Výrobní číslo. Další uživatelsky editovatelné okno jsou Měřicí přístroje, zde uživatel zadává měřené přístroje, jednotlivé přístroje oddělujeme klávesou Enter. Vepsané údaje ve formuláři se překopírují při generování na příslušné místo do reportu.

Aplikace je naprogramovaná tak, aby měl uživatel možnost vyobrazit report dvěma způsoby.



Obr. 5.1.22 Ukázka ukládání protokolu ve Front Panelu

A to přímým tiskem na defaultně nastavenou tiskárnu (v MS Windows®), nebo uložením do souboru s příponou HTML. Aplikace je přednastavena tak, aby výchozí hodnota byla nastavena na přímý tisk – *Tiskárna*. Tlačítkem TISK spustíme tisk reportu. Druhá možnost exportování reportu je jeho uložení do souboru s příponou HTML. Zvolíme kurzorem myši volbu HTML a klikneme na tlačítko se symbolem složky pro nastavení cílového adresáře vyexportovaného souboru. V programu je již přednastavena přípona souboru na \*.html, tudíž uživatel musí vyplnit pouze název, pod jakým chce soubor uložit a potvrdit tlačítkem OK. Dialogové okno je stejné jako u MS Windows®. Tímto je nastavena cesta pro uložení souboru, stisknutím tlačítka TISK dojde k uložení souboru do složky nastavené výše. Pokud by uživatel nezadal cestu pro uložení a název souboru, je cesta nastavena na adresář TEMP ve Windows, to zajišťuje v programu blok Temporary Directory, který zajistí uložení na jakémkoliv počítači s operačním systémem Windows do stejné složky, nezávisle na množství uživatelských profilů. Název uloženého souboru bude *mereni\_ss\_motor.html*, každým uložením je soubor přepisován, bez dotázání, na novější verzi. Pokud bychom chtěli vytvořit export do souboru typu PDF, stačilo by nastavit ve Windows jakoukoliv virtuální pdf tiskárnu (např. PDFCreator) jako výchozí a po stisknutí tlačítka TISK nastaveného na přímý tisk na tiskárnu se nám vytvoří PDF soubor.

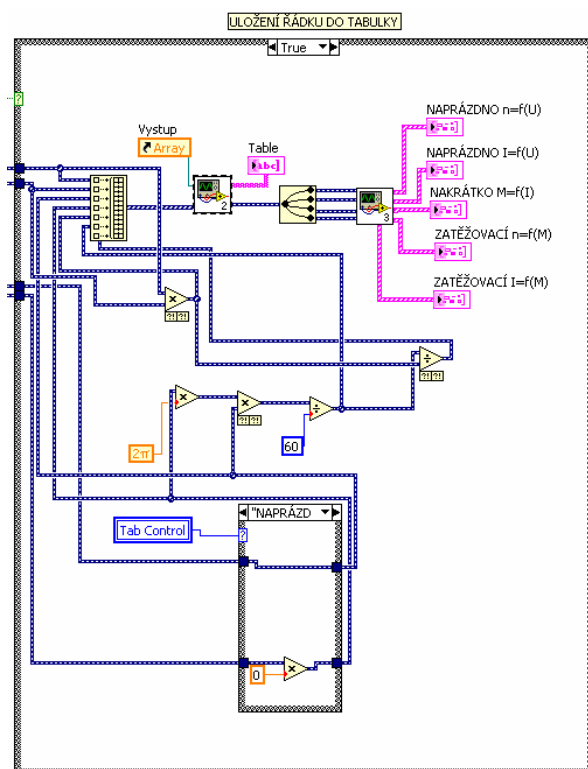


Při programování reportu jsme používali „programátorský“ způsob zápisu, používaný například při tvorbě webových stránek, tzv. tagů. Tag je značka vymezující vzhled a smysl dokumentu, zobrazuje se v ostrých závorkách. Pro ukázkou je níže zobrazena tabulka námi použitých párových i nepárových tagů.

<big>	zvětšení písma o jeden stupeň
<small>	zmenšení písma o jeden stupeň
<b>	tučný text
<tab>	nový odstavec
<newline>	nový řádek

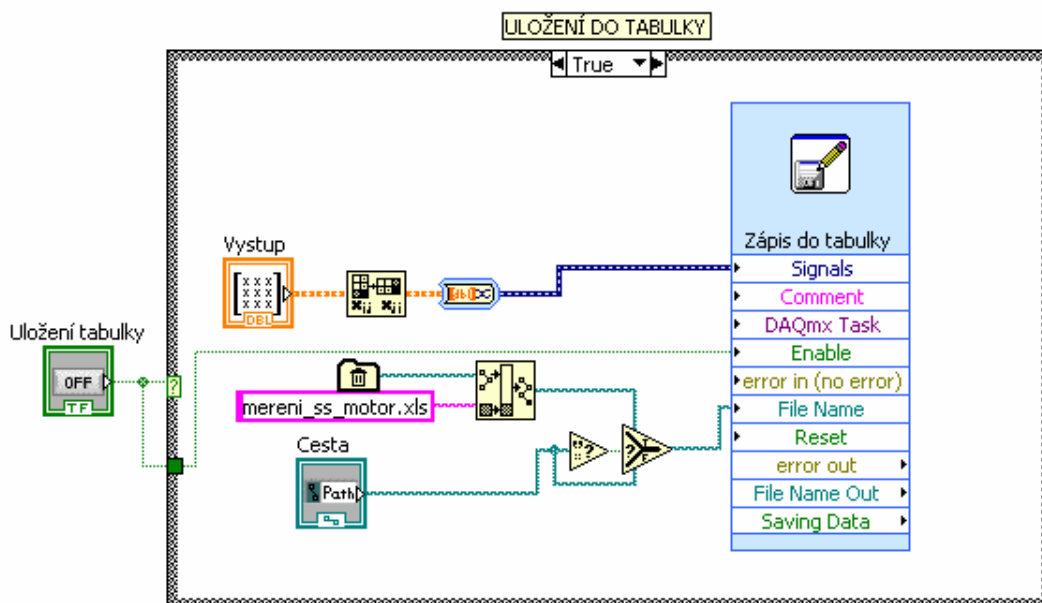
### 5.2.5 Vytvoření tabulky

Snímané signály přivedeme na blok Build Array, který zřetězí vícenásobná pole nebo přidá elementy k n-rozměrnému poli. Přivádíme dohromady sedm signálů, čtyři měřené a tři vypočtené. Vypočtené signály počítáme kontinuálně ze snímaných signálů pomocí numerických funkcí Add a Multiply. Z Build Array vedeme wire do bloku Insert Into Array, který vytvoří n-dimenzionální proměnnou, kterou zapíše do matice, matici transponujeme a převedeme její výstup na dynamická data. Ta vstupují do bloku Build Table, který nám signál převede do grafické podoby tabulky. Aby zápis dat do tabulky neprobíhal spojitě, úměrně se smplovací frekvencí (docházelo by ke kontinuálnímu zápisu měřených hodnot řádově v desítkách za sekundu), ale pouze v čase, který si uživatel zvolí stisknutím tlačítka. Program je navržen tak, aby při stisknutí tlačítka *Uložit řádek do tabulky* se uložily pouze hodnoty do jednoho řádku v čase stisknutí tlačítka. Tato vlastnost je realizována pomocí Case Structure, v které je implementováno celé řízení zápisu do matice a tedy i tabulky. Aby cyklus v Case Structure proběhl jen na jeden sample, je tato vlastnost ošetřena tlačítkem s funkcí *Latch when pressed* - sepne pouze na jeden cyklus. Pod tabulkou se nachází tlačítko Reset, které vymaže obsah tabulky a všech grafů. Tato funkce je realizovaná přes Invoke node a prvek Reinit To Default, kdy do Case Structure jsou zavedeny všechny Reinity a přes tlačítko RESET jsou vynulovány na výchozí hodnoty.



Obr. 5.1.23 Část programu zajišťující ukládání řádku do tabulky

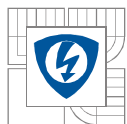
Pro další analýzu naměřených a vypočtených hodnot je vytvořen export do souboru typu XLS.



Obr. 5.1.24 Uložení tabulky do Excelu

Zápis je proveden přes blok Write to Measurement file, kde hodnoty načítáme z matice, do které jsou ukládány řádky měřených hodnot, tuto matici transponujeme a převedeme na dynamická data, která zapíšeme do souboru.





Program LabVIEW neumí exportovat do souboru s příponou XLS, ale pouze do souboru s příponou \*.lvm, avšak pokud tomuto souboru vnutíme excelovskou příponu, dokáže ji akceptovat. Sice při otevření vyexportovaného souboru v programu MS Excel se objeví hláška, že soubor není vytvořen v programu Excel, nebo může být poškozen, chceme-li opravdu soubor otevřít, klikneme na ANO a můžeme editovat naměřené hodnoty. Zápis do Excelu je vytvořen pro další zpracování naměřených hodnot, např. pro vytváření vlastních závislostí a charakteristik. Dalším důvodem exportování do Excelu je, že v tabulce na Front Panelu je přesnost zobrazování nastavena na dvě desetinná místa (zaokrouhlením), je to z důvodu přehlednosti a také omezeným místem, který nám poskytuje pracovní plocha o rozlišení 1024x768. Kdežto ve vyexportovaném souboru do Excelu je přesnost zobrazování nastavena na šest desetinných míst.

Parametry ukládání jsou nastaveny tak, aby při uložení tabulky o stejném jméně došlo k jejímu přepsání, do souboru se ukládají pouze data bez časových údajů, a to v pořadí v jakém jsou zobrazeny ve Front Panelu. Cestu pro uložení souboru si opět volí sám uživatel, pokud tak neučiní, bude vybrána složka TEMP ve Windows, stejně jako u reportu viz. výše v textu.

### 5.2.6 Grafy

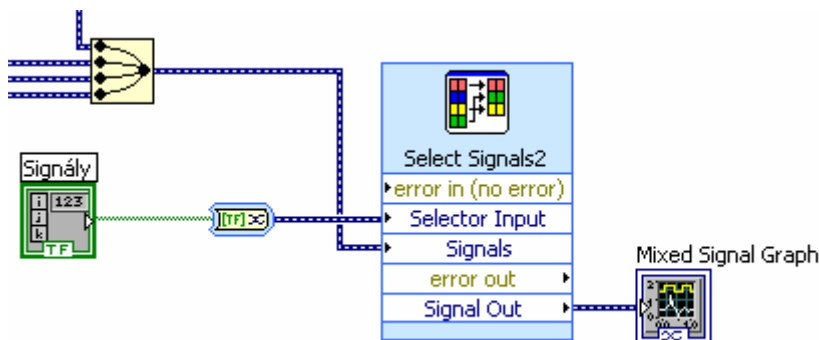
Vykreslení průběhu do grafů je vytaženo ze stejného nodu jako při tvorbě tabulky. Opět je třeba transponovat matici a převést ji na dynamická data a ty rozdělit blokem Split Signal na jednotlivé dílčí signály. Blok Split Signal dodržuje pořadí signálů v takové souslednosti, v jaké jsme jej připojili na Build Array. Graf je vykreslován z naměřených hodnot zapsaných v tabulce, nevykresluje se z průběžných! Ze Split Signálu jsme vybrali jednotlivé wire, které odpovídají měřeným veličinám pro určitý typ charakteristiky, např. pro charakteristiku nakrátko (závislost proudu na momentu). Zvolili jsme tyto wire: Druhý, který reprezentuje proud a připojili jsme jej na vstup osy X. A čtvrtý wire, který představuje moment, ten jsme připojili na Y osu. K výstupu Build Graphu jsme připojili zobrazovač XY Graph, který jsme provázali s Front panelem. Obdobně jsme postupovali pro každý graf zvlášť.

### 5.2.7 Osciloskop

V programu je jako první záložka nastaven osciloskop, který zobrazuje průběhy okamžitých hodnot. Abychom nemuseli pro každou měřenou veličinu vytvářet jeden samostatný zobrazovač signálu (waveform), vytvořili jsme v diagramu blok Select Signal, který dokáže propouštět jen vybraný druh signálů. Tímto řešením jsme ušetřili spoustu místa na pracovní ploše. Jako zobrazovač jsme nepoužili klasický waveform, ale jeho modifikovanou verzi Mixed Signal Graph, která dokáže zobrazit informace o měřeném signálu. Tato varianta nastavení osciloskopu má několik výhod, a to např. při hledání problému v zapojení, kdy můžeme editovat každý signál zvlášť. Samozřejmě pozorovat signály bychom mohli i bez bloku Select Signal, ale v jednom waveformu by se nám všechny signály zobrazovaly zároveň. Waveform má osy nastaveny na autoscale, tzn. osy



přizpůsobují své měřítko měřené veličině. Tudíž signály s nižší amplitudou by oproti těm z vyšší nebyly téměř viditelné. V praxi to znamená, že pokud bychom se chtěli podívat na průběh momentu, nebylo by to při vyšších otáčkách téměř možné, jelikož moment má maximální amplitudu řádově v jednotkách, kdežto otáčky v tisících jednotkách.



Obr. 5.1.25 Výběr signálů pro osciloskop

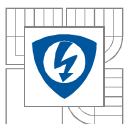
Vlevo v obrázku je vidět blok Merge Signal, který nám měřené signály z DAQ karty spojil v jeden víceúrovňový, ten přivádíme do bloku Select Signal. Pomocí booleovského tlačítka určíme jaký signál má být vybrán. Mezi tlačítky a blokem pro výběr signálu musíme data překonvertovat z Array na dynamická data. Výstupní signál je zobrazován v bloku Mixed Signal Graph, který jak je vidno ve Front Panelu, zobrazuje i informaci, o jaký signál se jedná.

## 5.2.8 Převod měřených veličin

Všechny měřené veličiny jsou snímány přes DAQ asistenta a blokem Split Signal rozděleny do jednotlivých wires. Každý signál má jiný převod mezi napětím a žádanou veličinou, některé signály je třeba ještě před zpracováním vynásobit numerickou konstantou. Napětí snímáme přes napěťový dělič, jelikož snímané napětí může být až 24V, ale měřicí karta dokáže snímat napětí pouze do  $\pm 10V$ , kde napěťový dělič je snižující v poměru 3:1. Proto jsme v Block Diagramu napěťový signál vynásobili třemi. Stejně tak převod otáčkoměru je 1000:1, tzn. že při 1000 otáčkách naměříme 1V, proto jsme signál s otáčkami vynásobili konstantou 1000. Obdobně jsme vytvořili konstantu u momentu, kdy převod tenzometru je 1,023 N.m/V. U proudu bylo řešení obtížnější, jelikož jako měřicí přístroj proudu jsme použili klešťový ampérmetr, který má přepínání rozsahu 10x nebo 100x. Tento parametr jsme v LabVIEW realizovali pomocí Case structure s výše uvedenými konstantami, přepínání je zajištěno booleanovským tlačítkem.

## 5.2.9 Vytvoření instalačního souboru

V diplomové práci je přiloženo CD s instalací námi vytvořeného programu. Aby bylo možné provozovat program EasyMot, musí být nainstalováno na PC LabVIEW nebo jeho Run-Time Engine.




Výhodou Run-Time Enginu je, že je volně šiřitelný a zcela zdarma. Proto jsme do instalátoru zakomponovali i NI LabVIEW Run-Time Engine 8.5. V této konfiguraci je možné spouštět program na kterémkoliv PC, samozřejmě musí splňovat výše uvedené požadavky na hardware.

Na CD je složka EasyMot, která obsahuje spustitelný instalační soubor setup.exe. Po spuštění tohoto souboru se spustí instalátor a dotáže se na dvě cesty pro uložení. První je pro uložení samostatného programu EasyMot a druhá je pro uložení emulátoru LabVIEW.


Program EasyMot není určen ke komerčnímu užití! Byl vytvořen v LabVIEW 8.5 licencovaném na školní verzi pro VUT v Brně. Proto nemůže být program využit za účelem obohacení. Program je volně šiřitelný.

## 5.3 Postup měření


### 1. Spuštění programu


V nabídce start ve Windows spustíme zástupce programu EASYMOT, zobrazí se měřicí panel. Stiskneme tlačítko Run Continously , tímto je program spuštěn a připraven k měření.

### 2. Spuštění měření

Stiskem tlačítka I/O  aktivujeme měřicí kartu a můžeme pozorovat měřené veličiny na osciloskopu.

### 3. Aktivace zdroje

Přepínačem Heinzinger navážeme komunikaci s programovatelným zdrojem Heinzinger. Pokud budeme používat jiný zdroj napětí než Heinzinger PCYs 24-300, třeba novější Heinzinger, nebo jiný zdroj napětí pracující na protokolu SCPI s příkazy VOLT\_#, CURR\_#, atd., je pod tlačítkem RESET zdroje dialogové okno s názvem GPIB Adresa, do kterého stačí vepsat číslo GPIB adresy, na kterém daný zdroj komunikuje, stisknutím tlačítka  je potvrzeno spojení.

*Pozn.: Pro nouzové vypnutí zdroje slouží tlačítko RESET zdroje . Toto tlačítko vynuluje táhla napětí a proudového omezení a do zdroje Heinzinger pošle pomocí SCPI protokolu příkaz pro jeho resetování do výchozích (nulových) hodnot.*

### 4. Nastavení ochrany motoru

Pod záložkou UČITEL nastavíme maximální hodnotu proudového omezení. Tato hodnota proudu bude maximální, která bude moci motorem protéct. Pokud bude uživatel programu chtít pomocí posuvníku proudového omezení nastavit vyšší hodnotu, než bude nastavena jako maximální, bude program jeho požadavek ignorovat a rozsvítí diodu POZOR!

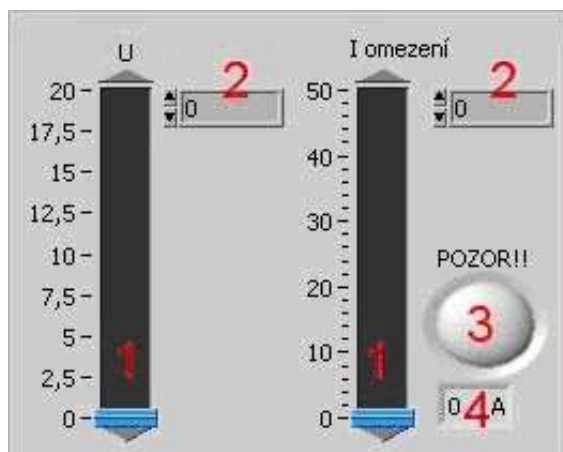
Dioda zhasne po snížení hodnoty proudového omezení pod maximální úroveň. Toto opatření je navrženo jako ochrana před zničením měřeného motoru.

### 5. Vyplnění údajů o měřeném přístroji

V záložce VSTUPNÍ DATA vyplníme jednotlivé kolonky. Mezi kolonkami se můžeme přepínat tabulátorem.

### 6. Ovládání motoru

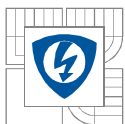
Po nastavení proudového omezení můžeme postupně zvyšovat hodnotu napětí a tím roztáčet motor.



1. šoupátka pro nastavení hodnoty napětí nebo I omezení
2. velikost nastavované hodnoty
3. kontrolka upozorňující na překročení dovolené hodnoty
4. aktuální hodnota proudového omezení ve zdroji

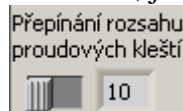
### 7. Zobrazení signálů

Pod záložkou OSCILOSCOP vidíme obrazovku osciloskopu s aktuálně zobrazovanými signály. Nalevo od obrazovky jsou čtyři tlačítka pro zapínání a vypínání zobrazování signálů. Tlačítka jsou pod sebou v pořadí: napětí U, proud I, otáčky n, moment M. Tímto je možné na osciloskopu sledovat pouze jednu žádanou veličinu. Osy osciloskopu jsou nastaveny na autoscale a budou se dynamicky přizpůsobovat zobrazovanému signálu.



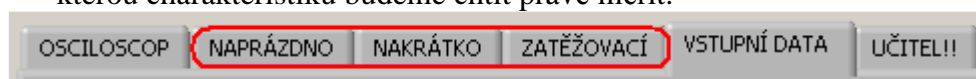
## 8. Nastavení proudového rozsahu

Pro správné měření musíme pomocí přepínače (viz. Obr.) zvolit stejnou konstantu, jaká je na proudových kleštích.



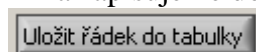
## 9. Výběr druhu charakteristiky

Dále vybereme jednu z červeně označených záložek viz. Obr., podle toho, kterou charakteristiku budeme chtít právě měřit.



## 10. Zapisování měřených hodnot do tabulky

Nastavujeme napětí podle toho, jaký druh charakteristiky chceme měřit a zapisujeme do tabulky. Do tabulky zapisujeme pomocí tlačítka.

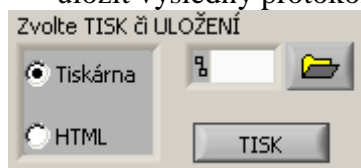


*Pozn.: Program je nastaven, aby při měření charakteristiky naprázdno byl měřený moment vždy roven nule. Tento požadavek je realizován tak, pokud má uživatel aktivní záložku NAPRÁZDNO, jsou hodnoty momentu násobeny nulou. Tento požadavek vyplývá z ideální charakteristiky naprázdno, kdy motor se otáčí zcela nezatížený.*

*Stejný požadavek je i na charakteristiku nakrátko, kdy musí být nulové hodnoty u otáček. Opět pokud bude aktivní záložka NAKRÁTKO, budou hodnoty otáček násobeny nulou. Vycházíme z charakteristiky nakrátko, kdy hřídel motoru je zablokována.*

## 11. Vytisknutí výsledného protokolu

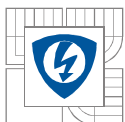
Jakmile máme požadovaný počet hodnot, můžeme si vytisknout a nebo uložit výsledný protokol.



Aplikace je přednastavena tak, aby výchozí hodnota byla nastavena na přímý tisk – *Tiskárna*. Tlačítkem TISK spustíme tisk reportu. Druhá možnost exportování reportu je jeho uložení do souboru s příponou HTML. Zvolíme kurzorem myši volbu HTML a klikneme na tlačítko se symbolem složky pro nastavení cílového adresáře vyexportovaného souboru. V programu je již přednastavena přípona souboru na *\*.html*, tudíž uživatel musí vyplnit pouze název, pod jakým chce soubor uložit a potvrdit tlačítkem OK.

## 12. Uložení do Excelu

Pokud budeme chtít dále zpracovávat naměřené hodnoty, můžeme si tabulku naměřených hodnot vyexportovat do Excelu.



A to pomocí tlačítka

Uložit do \*.XLS

Nejdříve ale musíme nastavit cílový adresář pro uložení souboru, viz. bod výše.

### 13. Vynulování hodnot

K vynulování tabulky a grafů slouží tlačítko Reset

Reset

Vynuluje veškeré naměřené hodnoty výjma vyexportovaných uložených do souboru. Použitím tlačítka se nezruší nastavená hodnota proudového omezení v záložce UČITEL a zapsaná data v záložce VSTUPNÍ DATA. Toto tlačítko musí být také použito při každém novém měření, tzn. při měření nové charakteristiky. Pokud uživatel nevyresetuje hodnoty před novým měřením, nebude měření správné.

Předchozí postup je popsán pro měření jednoho druhu charakteristiky, např. charakteristiky naprázdno. Pro zbývající dvě charakteristiky, nakrátko a zatěžovací, je postup stejný. Nastavení prvních osmi bodů zůstane zachováno, takže uživatel může pokračovat od bodu devět.

## 5.4 Použité přístroje v měřicím systému:

Dynamometr MEZ Náchod

otáčky	n	3000	min <sup>-1</sup>
výkon	P	550	W
moment	M	1,75	N.m
napětí buzení	U <sub>b</sub>	220	V
max.budící proud	I <sub>bmax</sub>	2	A
napětí kotvy	U <sub>a</sub>	220	V
max.proud kotvy	I <sub>a</sub> <sub>max</sub>	4	A
Napájení buzení dynamometru:			
elektronický stejnosměrný zdroj		24V / 2A	

Snímač momentu:

tenzometr INTERFACE SM 25

převod 1,023 N.m / V

Snímač otáček:

optický Metrix

převod 1000 min<sup>-1</sup> / V

Snímání proudu:

Klešťový ampérmetr – Chauvin Arnooux E3N

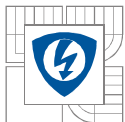
převod 100mV – 10mV/A

Měřený motor:

Magnetron S230

v.č. 443111574050

12V 180W 1400 min<sup>-1</sup>



## 6 ZHODNOCENÍ

Tato práce je věnována virtuálním laboratořím, pojednává o jejich využití v praxi. V prvním bodě je představen pojem virtuální laboratoř a rozdělení na dva základní druhy.

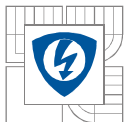
Druhá kapitola je věnována teorii stejnosměrných strojů, zvláště stejnosměrnému motoru se sériovým buzením, kde jsou vysvětleny základní principy a vlastnosti stroje.

Dále je představen program LabVIEW, jeho struktura, využití, základní vlastnosti. LabVIEW je vývojové grafické prostředí zaměřené na vývoj aplikací umožňující měření, sběr a analýzu dat.

V poslední části diplomové práce je popsána realizace námi vytvořeného programu EasyMot. Program byl vytvořen pro měření stejnosměrného sériového motoru. Pomocí programu EasyMot může uživatel plně ovládat motor a snímat jeho veličiny přes program v počítači. Tato skutečnost urychluje analýzu a měření motoru, ale také díky řízení přes osobní počítač vytváří bezpečnější prostředí pro uživatele a eliminuje možnost doteku osoby se živými částmi měřené soustavy. Program je navržen tak, aby bylo možné pouze s malými úpravami měřit i jiné druhy motorů, asynchronní, synchronní. Spíše než softwarově jsme omezeni hardwarově, zejména bychom potřebovali programovatelné střídavé zdroje napětí, s komunikačním rozhraním GPIB. Tímto by se mnohem lépe využil potenciál automatizované laboratoře a program by měl mnohem širší možnosti uplatnění.

Prostředí programu je uživatelsky příjemné a ovládání je intuitivní. Program má v sobě implementován i ochranný prvek, který znemožňuje náhodné zničení motoru, či měřicí soustavy. Program kontinuálně vypočítává hodnoty z naměřených veličin a dokáže vytvořit výstupní protokol s naměřenými veličinami a grafy. Program by bylo možné dále rozšířit o dálkové řízení přes internet, což by umožnilo měření daleko většímu počtu uživatelů. S realizací měření přes internet se nabízí myšlenka vytvoření jazykových mutací, což v programovém prostředí LabVIEW není problém.

EasyMot je možno používat jako podporu laboratorních cvičení v ústavu výkonové elektrotechniky a elektroniky.



## LITERATURA

- [1] Bašta J., Chládek J., Mayer I.: Teorie elektrických strojů, SNTL Praha 1968
- [2] Cigánek L., Stavba elektrických strojů, SNTL, Praha 1958
- [3] Čejka M., Použití počítačů v měřicí technice, VUT Brno 2002
- [4] <http://www.ni.com/labview>
- [5] <http://www.fzu.cz/departments/multilayer/labview/obsah.html>
- [6] Kunát J., Automatizované měřicí systémy pro elektrické stroje
- [7] <http://www.home.agilent.com>